

Utilizing Machine Learning Methods for Usability Evaluation in Learning Management Systems

Richard A. Torres Molina

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Application

Mohammed S. Seyam, Chair

Edward A. Fox

D. Scott McCrickard

April 15, 2024

Blacksburg, Virginia

Keywords: usability evaluation, machine learning, learning management systems

Copyright 2024, Richard A. Torres Molina

Utilizing Machine Learning Methods for Usability Evaluation in Learning Management Systems

Richard A. Torres Molina

(ABSTRACT)

The concept of usability refers to a user's capability to interact with a system to fulfill goals in terms of task completion (effectiveness), time measurement (efficiency), and positive attitude (satisfaction). The strategy for usability evaluation in software systems usually involves questionnaires, user testing, and heuristics. Although these methods have been widely used due to several benefits, there are challenges related to time consumption and embedded bias. In response to these challenges, this work proposes a hybrid approach based on usability questionnaire answers and machine learning algorithms to predict usability scores. We describe three different experiments with features extracted from a Learning Management System. These features were applied in the Machine Learning algorithms Linear Regression, Decision Trees, Random Forest, and Neural Networks in three experiments. Random Forest produces the best performance of average mean square error and root mean square error among machine learning algorithms. The results are promising, though there are alternatives for improvements for better performance of the System Usability Scale and UseLearn scores prediction. This approach has potential as a reliable predictive tool for usability scores, which would help create software systems that better satisfy users' needs.

Utilizing Machine Learning Methods for Usability Evaluation in Learning Management Systems

Richard A. Torres Molina

(GENERAL AUDIENCE ABSTRACT)

Instructors and students have used online platforms known as Learning Management Systems (LMSs) to improve learning and satisfaction. Students need to achieve their learning goals by interacting with these systems. To achieve these goals, usability evaluation involves ensuring that LMSs attain effectiveness (task completion), efficiency (time measurement), and satisfaction (positive attitude). Usability evaluation usually follows questionnaires, user testing of the LMS, and expert reviews. Although these methods are widely used due to several benefits, they face challenges related to trying these software systems multiple times until the system satisfies student needs and human subjectivity perception. To face these challenges, promote student engagement with the system, and create a better design in the LMS courses, we propose a hybrid approach based on data, questionnaire answers, and machine learning algorithms to predict usability scores. We evaluated this approach through a case study with data collected from undergraduate students at Virginia Tech. The results showed different advantages and drawbacks of machine learning performance. The approach contributes to the engineering and computing education field by providing a reliable predictive tool for usability scores to improve the student learning experience and the features of the LMS.

Dedication

To the divine energy and my mother

Acknowledgments

I thank my thesis advisor Dr. Mohammed Seyam for supporting my ideas, writing feedback, and for the scholarship at Richard Tapia 2023. I thank Dr. Edward Fox for his academic advice, spiritual Reiki guidance, and for being part of my committee. I thank Dr. Scott McCrickard as my committee member. I acknowledge the seven faculty members and seventy students for their support in collecting the dataset. I am grateful to Dr. Clifford Shaffer for my admission and teaching assistantship deferral. I want to thank Fulbright Ecuador for their guidance and financial support during the application process to study in the United States. Additionally, I would like to acknowledge my past mentors and organizations who supported me during my college years (Ecuador, Colombia, Spain, Venezuela, the Czech Republic, Italy, China, the United Kingdom, Brazil, France, and Ireland). Finally, I am forever grateful to my mother, and God for allowing me to understand humanity a little more, and myself.

Contents

- List of Figures** **ix**

- List of Tables** **xi**

- 1 Introduction** **1**
 - 1.1 Motivation 1
 - 1.2 Research Questions 2
 - 1.3 Overview 3

- 2 Literature review** **4**
 - 2.1 Usability 4
 - 2.2 Usability Evaluation Methods 4
 - 2.2.1 Questionnaires 6
 - 2.2.2 User Testing 8
 - 2.2.3 Heuristics 10
 - 2.2.4 Log analysis 11
 - 2.2.5 UE methods combination 11
 - 2.3 Machine Learning 11
 - 2.3.1 ML Techniques in Usability Evaluation for Software Systems 12

2.3.2	ML workflow	13
2.3.3	ML embedded in usability evaluation	14
3	Methodology	17
3.1	Methods	17
3.1.1	Moodle course	17
3.1.2	Subjective approach	23
3.1.3	Objective approach	26
3.2	Experimental Set-Up	31
3.2.1	Environment	31
3.2.2	Data collection	31
3.2.3	Data pre-processing	32
3.2.4	Supervised Machine Learning algorithms	38
3.2.5	Evaluation Metrics	39
4	Results	40
4.1	Dataset statistics	40
4.2	Experiments	41
4.2.1	Experiments with 6 features	42
4.2.2	Experiments with 7 features	45
4.2.3	Experiments with 10 features	47

5	Discussion	51
6	Conclusion and Future work	55
6.1	Conclusion	55
6.2	Future work	56
	Bibliography	57
	Appendices	67
	Appendix A User study documents	68
A.1	Questionnaires	68
A.1.1	System Usability Scale (5-point Likert scale chart)	68
A.1.2	UseLearn checklist (5-point Likert scale chart)	69
A.2	Recruitment strategies	70
A.2.1	Email sent to professors	70
A.2.2	Student Google Forms	73
A.2.3	Student Flyer	77
A.2.4	Institutional Review Board (IRB) Confirmation Approval	78
A.2.5	Student email confirmation study	79

List of Figures

2.1	Usability Evaluation Methods Classification [1].	5
3.1	Moodle online course.	18
3.2	Steps 1-3 to finish Quiz. (a) Step 1. Choosing the “Quiz” tab from the left bar from the course, (b) Step 2. Click on “Attempt quiz now”, (c) Step 3. Click on “Start attempt”.	19
3.3	Steps 4-6 to finish Quiz. (a) Step 4. Select a “radio button” as an answer, (b) Step 5. Click “Next page” until the 10 questions are answered, (c) Step 6. Click on “Submit all and finish”.	20
3.4	Steps 1-4 to post into a forum. (a) Steps 1-2. Choosing the “Discussion Forum” from the left bar from the course and click on “Forum”, (b) Step 3. Click on “Reply” and write the message, (c) Step 4. Click on “Post to forum”.	22
3.5	Steps 1-5 to finish the Assignment. (a) Steps 1-2. Choosing the “Assignment” from the left bar from the course and click on “Add submission”, (b) Step 3. Click on the arrow to upload the file, (c) Steps 4-5. Click on “Choose file” and “Upload this file”.	24
3.6	Steps 6-8 to finish the Assignment. (a) Step 6. Click on “Save changes”, (b) Step 7. Click on “Submit assignment”, (c) Step 8. Click on “Continue”.	25
3.7	Log analysis representation [2].	27
3.8	Linear regression representation [3].	28

3.9	Decision tree representation [4].	29
3.10	Random forest representation [4].	30
3.11	Neuron unit representation [3].	30
3.12	Multilayer artificial neural network (ANN) representation [4].	31
3.13	Database overview.	34
3.14	Features distribution.	37
4.1	MSE from the usability scores with different Machine Learning Algorithms with 6 features selected.	43
4.2	RMSE from different Machine Learning Algorithms with 6 features selected.	45
4.3	MSE from different Machine Learning Algorithms with 7 features selected. .	46
4.4	RMSE from different Machine Learning Algorithms with 7 features selected.	47
4.5	MSE from different Machine Learning Algorithms with 10 features selected.	49
4.6	RMSE from different Machine Learning Algorithms with 10 features selected.	50

List of Tables

3.1	Steps to finish the quiz.	21
3.2	Steps to finish posting into the discussion forum.	21
3.3	Steps to finish assignment.	23
3.4	Exclusion criteria.	33
3.5	Features description.	33
3.6	Components action and target description.	35
3.7	Target description.	36
3.8	Features description.	38
3.9	Parameter settings for ML algorithms on grid search algorithm.	39
4.1	Features mean and standard deviation.	40
4.2	Target mean and standard deviation.	41
4.3	Parameter settings for ML algorithms.	42
4.4	MSE from different Machine Learning Algorithms with 6 features selected.	43
4.5	RMSE from different Machine Learning Algorithms with 6 features selected.	44
4.6	Parameter settings for ML algorithms.	45
4.7	MSE from different Machine Learning Algorithms with 7 features selected.	46
4.8	RMSE from different Machine Learning Algorithms with 7 features selected.	47

4.9	Parameter settings for ML algorithms.	48
4.10	MSE from different Machine Learning Algorithms with 10 features selected.	48
4.11	RMSE from different Machine Learning Algorithms with 10 features selected.	48

List of Abbreviations

ACTES Automated Clinical Trial Eligibility Screener

AI Artificial Intelligence

ASD Autism Spectrum Disorder

ASQ After-Scenario Questionnaire

CAMLS Context-Aware Mobile Learning System

CART Classification and Regression Tree

CF Consistency and Functionality

CM Course Management

CSS Cascading Style Sheets

CSUQ Computer System Usability Questionnaire

CT Computed Tomography

CTk Cyclic Task

DOM Document Object Model

EHR Electronic Health Record

EP Error Prevention

GUI Graphical User Interface

HTML HyperText Markup Language

IRB Institutional Review Board

ISO International Organization for Standardization

LIME Local Interpretable Model Agnostic Explanations

LMS Learning Management System

LT Laborious Task

MAF Missing Action Feedback

ML Machine Learning

MLP Multilayer Perceptron

MOOC Massive Open Online Course

MRI Magnetic Resonance Imaging

MSE Mean Square Error

MUsE Model Usability Evaluation

NL Natural Language Processing

NN Neural Network

OMP Observational Medical Outcomes Partnership

PSSUQ Post-Study System Usability Questionnaire

RMSE Root Mean Square Error

RoB Risk of Bias

SR Systematic Review

SUS System Usability Scale

SVM Support Vector Machine

UE Usability Evaluation

UEQ User Experience Questionnaire

UEt Undescriptive Element

UI User Interface

UMUX-Lite Usability Metric for User Experience Lite

URL Uniform Resource Locator

USE Usefulness, Satisfaction, and Ease of Use

UUX Usability-User Experience

UX User Experience

VUS Voice Usability Scale

xAI explainable Artificial Intelligence

Chapter 1

Introduction

This thesis presents a hybrid usability evaluation approach based on user logging data, quantitative usability questionnaire answers, and Machine Learning (ML) techniques to predict usability scores for a Learning Management System (LMS). The approach provides a novel strategy as a predictive tool for usability scores, which would help the research community create LMSs that satisfy users' needs.

1.1 Motivation

Usability evaluation (UE) is crucial for the quality of software systems to promote user goal achievement based on effectiveness, efficiency, and satisfaction. The subjective methods are the current methodology for UE: questionnaires, user testing, heuristics, and focus groups. This evaluation helps find usability issues. However, they present gaps for improvement. These gaps are the time needed to test a software system multiple times until the system satisfies user needs and human subjectivity perception. Also, they lack the capability for real-time configurations to customize different software systems. An objective quantitative analysis is required to provide better software systems that satisfy user needs in the long term. This analysis is a narrow research area with user logging data and ML. However, there is no standard automatic objective identification tool for UE with ML where the score predictions will help the software developers and User Experience (UX) designers to improve Learning

Management System (LMS) software over time after corrections. A potential alternative is automatic detection tools.

There are automatic detection tools for UE but with a scarce overview of user log activity data and data-driven machine learning (ML) techniques. There are still open questions about ML embedded in UE. For example, which interactions (features) between a user and a software system are required to predict positive overall usability and which issues in software systems usability can be predicted with ML. The features selected for the ML algorithms in the previous state of the art are task duration, clicks, keystrokes, URL (Uniform Resource Locator), event type, and among others to classify usability issues or predict a general score. These features are extracted after the users finish certain tasks depending on the software system. UE is crucial to creating software systems that fulfill user goals. In the educational field, LMS is a software system used by students daily. UE should be considered to create better learning experiences. Therefore, this thesis shows a hybrid approach based on user logging data, usability questionnaire answers, and ML techniques to predict usability scores for the Moodle LMS. This approach provides a potential strategy for UE, which would help researchers, software developers, and UX designers create LMS software systems that satisfy their users' needs.

1.2 Research Questions

In this thesis, I present an alternative UE methodology for usability score prediction based on user interaction with a LMS software system, usability questionnaires, and ML algorithms. The two research questions are:

1. Are supervised machine learning algorithms able to predict usability scores in LMSs?

2. What features should be employed to predict usability scores in LMSs?

1.3 Overview

The thesis is structured as follows: Chapter 2 presents the literature review about the intersection between usability evaluation and ML in software systems to show the current state of the art and potential research directions. Chapter 3 explains the methods for automatic usability evaluation of a LMS based on user logging data, questionnaires, and ML. Chapter 4 details the case study results with the ML algorithms. Chapter 5 discusses the results. Chapter 6 presents conclusions and future work.

Chapter 2

Literature review

2.1 Usability

Usability is an essential part of human-computer interaction research and a relevant component for quality User Experience (UX) [5]. The International Organization for Standardization (ISO) established the usability definition in ISO 9241-11 [6]. The concept refers to user goals achievement with a product considering different factors: task completion (effectiveness), time measurement (efficiency), and positive attitude (satisfaction). Jakob Nielsen [7] extended these factors to easy to learn (learnability), easy to remember (memorability), and low error rate (errors). Considering this perspective, usability evaluation (UE) methods have emerged to evaluate the usability of software systems to guarantee quality [8].

2.2 Usability Evaluation Methods ¹

Usability evaluation (UE) uses different methods and the complexity increases when selecting the best strategy for a specific software system. According to Andreas et al. [10], usability evaluation methods are classified into inspection (experts) and test methods (users). Figure 2.1 depicts a broader classification, where usability evaluation methods are divided into user testing, inspection, inquiry, analysis modeling, and simulation [1]. User testing methods are

¹Content extracted from our paper [9].

think-aloud, question-asking protocol, performance measurement, log analysis, and remote testing. Inspection methods are heuristic evaluation, cognitive walkthrough, perspective-based inspection, and guideline review. Inquiry methods are questionnaires, interviews, and focus groups. Analytic modeling involves cognitive tasks analysis, task environment analysis, and rules analysis. Simulation methods utilize agents or algorithms to replicate user behavior. All the methods converge and apply in multiple ways according to the context of the usability evaluation in software systems.

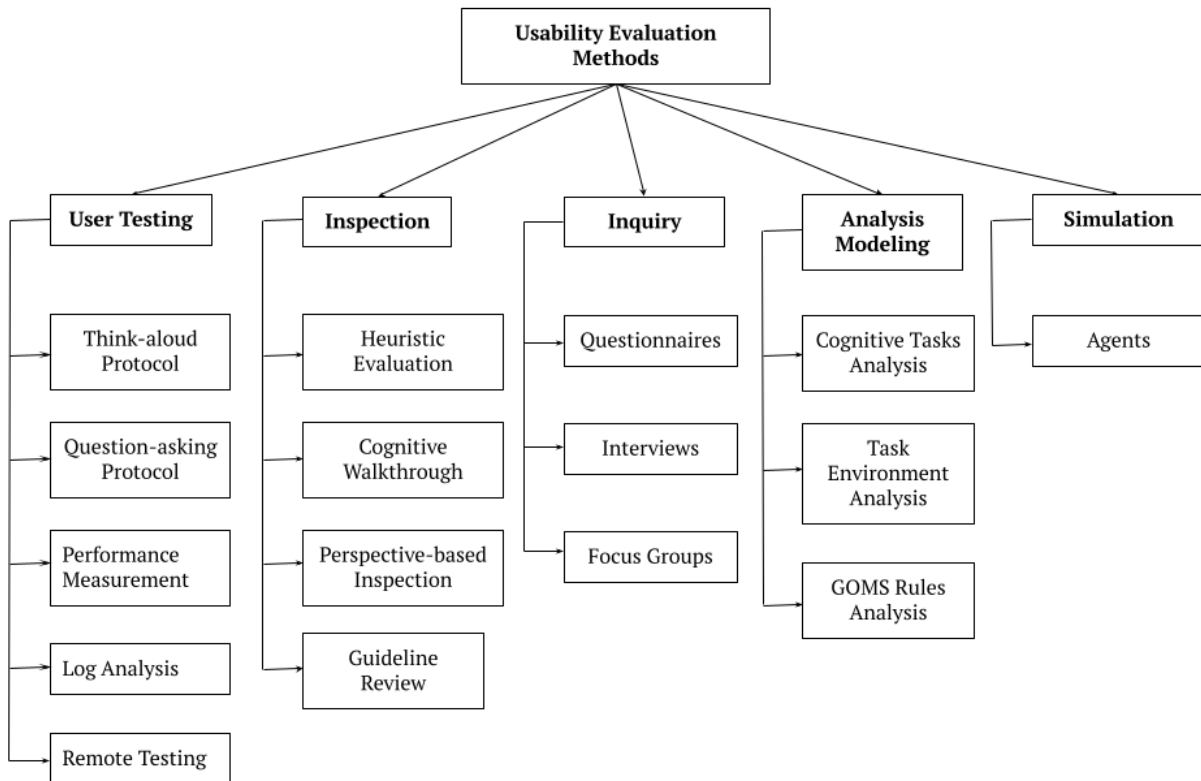


Figure 2.1: Usability Evaluation Methods Classification [1].

The UE methods can be classified as subjective and objective. The subjective methods are the current methodology for UE: questionnaires [11], user testing [12], heuristics [13], focus groups [14], and among others. Questionnaires include questions that measure usability. User

testing takes a list of pre-defined tasks on the screen, touch, or voice interaction between the user and the system. Heuristics are a set of guidelines where experts evaluate software's usability. Meanwhile, the focus group is an open discussion to evaluate potential usability issues in the platform with a small sample size. Although these UE methods have been broadly utilized, there are challenges related to embedded bias, time investment, and lack software systems adaptability after live updates. A potential alternative is an objective evaluation limited to logging user data, analytical tools, and Machine Learning (ML).

2.2.1 Questionnaires

Questionnaires include a set of questions measuring the usability of a software system. The questionnaires [11] are usually performed as pre-tests and post-tests to obtain information about the user and the interaction with the software system. It is recommended to apply standard questionnaires for reliability purposes. These questions' answers can be quantitative or qualitative. It is quantitative when the answers are on Likert scales. It is qualitative when the questions have open answers.

The standard questionnaires are SUS (System Usability Scale) [15, 16, 17, 18], After-Scenario Questionnaire (ASQ), Post-Study System Usability Questionnaire (PSSUQ) [19], User Experience Questionnaire (UEQ) [17], USE (Usefulness, Satisfaction, and Ease of use) [20, 21], Computer System Usability Questionnaire (CSUQ) [22], Usability Metric for User Experience Lite (UMUX-Lite) [23], UseLearn checklist [24], among others. Non-standard questionnaires have been created to address researcher's needs for different studies such as the e-learning platform Blackboard [25] on user student experience, UniStudium [26] (e-learning interface for learning analytics), Context-Aware Mobile Learning System (CAMLs) [27], academic websites [28], and an e-learning platform [29] for a programming course. Modern question-

naires are Model Usability Evaluation (MUSE) [30] and Voice Usability Scale (VUS) [31]. Usability evaluation in software systems research - including ML systems - is mostly with the questionnaire SUS [32, 33, 34, 35, 36, 37, 38]. SUS has been applied in the software industry with more interest by the research community in healthcare and education.

SUS in healthcare has been applied in different software products such as 1) relapse risk prediction in melanoma patients [32], 2) hand recognition system [33] for a surgical procedure, 3) Automated Clinical Trial Eligibility Screener (ACTES) [34], and 4) Autoscreen [35] for autism spectrum disorder (ASD) detection. The former is composed of the Observational Medical Outcomes Partnership (OMP) common data model independent of the electronic health record (EHR). The software user interface is composed of three main components: 1) input patient identification to load the data, 2) the patient data display, and 3) HTML (HyperText Markup Language) iframe with the user prediction. The second software system was implemented because computed tomography (CT) scans and magnetic resonance imaging (MRI) are crucial in the preoperative planning surgical plan. ACTES extracts patient demographics and clinical assessments from structured electronic health records (EHR) data and checks patients' conditions with unstructured narratives. The latter Autoscreen has a landing page, tutorial, screening activities, scoring form, and risk assessment.

In the education industry, SUS also has been applied in the following studies. Gates et al. [36] show the assessment of three software systems Abstractkr, DistillerSR, and RobotAnalyst in systematic reviews (SR). Each software system can predict the record's relevance. Another work was RobotReview [37], an open-source system that automatically extracts data from clinical trials for risk of bias in SRs. The RobotReviewer web interface has a PDF viewer and a summary of the automatic Cochrane Risk of Bias (RoB) tool. SUS was applicable in the adaptive learning system AdaptLearn [38]. This system is adaptive based on learner knowledge and learning style. SUS has been adopted in different e-learning systems [15, 16,

[17] as well as higher education levels [18]. Other questionnaires such as UseLearn checklist [24], USE [21], ASQ [19], PSSUQ [19], CSUQ [22], and UMUX-Lite [23] have been tested on adaptive [38], and non-adaptive platforms (Coursera, OpenLearning, Moodle, and Open Education).

Newly proposed questionnaires in usability evaluation are Model Usability Evaluation (MUsE) and Voice Usability Scale (VUS). MUsE [30] is a combination of “New ISO Standards for Usability, Usability Reports and Usability Measures” and “Usability Meanings and Interpretations in ISO Standards”. In this area, it was applied in the explainable artificial intelligence (xAI) framework LIME (Local Interpretable Model Agnostic Explanations). On the other hand, VUS [31] is an empirical approach with 10 questions on 7-point Likert scale for voice assistants - tested by Amazon Alexa. This was created because SUS is more suitable for graphical user interfaces (GUI) than voice assistants.

2.2.2 User Testing

User testing [12] is applied to users with a list of pre-defined tasks according to the software system. This testing [11] must consider real users without any relation with the expert or moderator to avoid bias. The task should be realistic, unambiguous, and time-specific to evaluate the system features. The testing is conducted anywhere, whether in a controlled environment or remotely with instructions [37]. A moderator presence is up to the researchers and the experiment design. There is an explanation about the test session to the users. Some experiments apply the thinking-aloud method where users express their thoughts while using the software system. The final results describe the issues and problems in the session(s).

User testing has been used in e-learning systems from various universities around the world [20, 28, 29]. User testing [15, 18, 38] takes place with different users following tasks according

to the software system. It is widely used in healthcare software systems. For instance, in the hand recognition software system [33], the physicians scrolled, moved, and zoomed the lumbar vertebra from computed tomography (CT). In the decision support tool [32], the physicians did queries and retrieved patient data. User testing goes to more domains like SR in education. Gates et al. [36] labeled each document as relevant or irrelevant in three software systems. User testing in a software system for a learning environment [39] is through e-learning platforms such as Learning Management Systems (LMSs). LMSs are electronic platforms with services such as course creation, course management, communication, and assessment like Moodle, Canvas, and Blackboard. Also, these systems require usability evaluation with users (students, faculty, and administrators) to promote their engagement and satisfaction. In AdaptLearn [40], the system assigned a module to study as a task (adaptive or non-adaptive condition) to undergraduate students. The participants on myCourseVille [19] follow a set of tasks such as downloading course materials, searching course schedules, setting up registration, and creating class schedules. To help students figure out solutions, usability testing is also done on the user interface (UI) [22] design on Massive Open Online Course (MOOC) platforms to promote user satisfaction [17, 23]. By bridging the gap between e-learning system design and usability evaluation, the major objective is to promote students' learning and fix usability problems.

Even user testing can be set up with voice commands in a voice assistant [31], where the participants follow a script with predefined tasks in Amazon Alexa, for example, checking the current weather and making an appointment with a doctor. User testing can be a self-reflection as seen in [30], where MUSE tackles the areas of effectiveness, efficiency, and satisfaction. In all these studies, the objective was to measure the usability of the software system. The usability results are found after the users answer the questionnaire questions. This helps find usability issues, and inconsistencies [34], or implement future work.

For instance, MUSE found that the connection between prediction probabilities and feature probability graph is incomplete (effectiveness); the tool is time-consuming (efficiency); and there is a lack of documentation and support (satisfaction). Another concern is that usability testing in voice assistants [31] is limited.

2.2.3 Heuristics

Heuristics [41, 42, 43, 44] are a set of guidelines where experts evaluate software's usability. The different guidelines are Nielsen's heuristics, Kujala's heuristics, Shneiderman's "Eight Golden Rules", Gerhardt-Powals' cognitive engineering principles, etc. The most popular heuristic is from Nielsen following the parameters: 1) Visibility of system status, 2) Match between system and the real world, 3) User control and freedom, 4) Consistency and standards, 5) Error prevention, 6) Recognition rather than recall, 7) Flexibility and efficiency of use, 8) Aesthetic and minimalist design, 9) Help users recognize, diagnose, and recover from errors, and 10) Help and documentation.

Heuristics have been employed in software systems like digital learning technology prototypes for monitoring intracranial pressure [45], the University of Hong Kong Libraries' mobile website [46], and an e-learning system [47] from a Nigerian case study. In this manner, heuristics [45] find usability issues in the e-learning system such as the color header and footer of the screen. Heuristics also detect [46] usability issues in regards to 1) consistency and standards, 2) flexibility and efficiency, and 3) a better approach for error recovery. The results [47] are capable of better improvement in student learning optimal support.

2.2.4 Log analysis

The objective evaluation is limited to user logging data and analytics tools. For example, Harrati et al. [15] collected user task movement in Moodle by estimation metrics such as clicks, task duration, cursor distance, and completion rate. Likewise, Fenu et al. [48] proposed an analytics tool for usability evaluation interface with specific factors such as session time and learnability score. There are automatic detection [49] tools for objective evaluation but with a limited overview of data-driven Machine Learning (ML) techniques. ML is a potential strategy to tackle usability evaluation research to help developers and UX designers for faster usability issue detection.

2.2.5 UE methods combination

Additionally, UE [50, 51, 52] can be a combination of questionnaires, user testing, interviews, focus groups, and log analysis with open gaps to explore. There is a lack of a unifying framework [52] in e-learning systems given UE methods. Scholars suggest e-learning systems focus on a specific engaged user, for a type of task with a specific type of technology. However, technological UE may not be able to account for a multidimensional and complex environment. The usability of e-learning technologies is more connected to higher educational contexts.

2.3 Machine Learning

Artificial Intelligence (AI) [3] is a universal field that incorporates characteristics from cognitive science, computer science, mathematics, and control theory. AI was born in 1956 during a summer workshop at Dartmouth College organized by John McCarthy. AI tries to

replicate human intelligence through intelligent agents. A subfield of AI is Machine Learning (ML). Tom Mitchell [53] defines it as a computer program that learns by accomplishing a specific task with a measure to improve its performance over time. ML is subdivided into 1) supervised learning, 2) unsupervised learning, and 3) reinforcement learning [54]. Supervised learning is based on a dataset with labeled data divided into training and testing, where performance is measured. Unsupervised learning does not require a training dataset, but a cluster is labeled by an expert. Reinforcement learning refers to an agent in an environment who learns through trial and error. In this process, a reward is calculated to learn a policy. ML [55] has different applications in these types of learning. Supervised learning is employed in spam filtering, handwriting recognition, face recognition, speech recognition, software engineering, and finance. Unsupervised learning is applicable in genomics, social network analysis, market segmentation, and computational biology. Reinforcement learning is handled in traffic signal control, gaming, and stock market analyses [56]. All the applications mentioned can employ a mixture of supervised, unsupervised, and reinforcement learning. In this context, all these applications demand software products in high scale demand. Thus, ML is applied in the software development cycle [54], with the most popular techniques like Neural Networks (NNs), random forests, decision trees, Support Vector Machines (SVMs), and naive Bayes [57].

2.3.1 ML Techniques in Usability Evaluation for Software Systems

Usability evaluation based on objective evaluation considers quantifiable metrics on software systems. The metrics are from user interaction with the system (clicks, session time, task time, etc.). In this context, the user is unaware of these interactions. These interactions and/or quantitative subjective answers from questionnaires are relevant to ML learning techniques for usability score predictions as an automatic detection tool. Therefore, this section

is an overview of a standard ML workflow and the work related to ML embedded in usability evaluation for software systems: e-learning systems, social networks, and user reviews.

2.3.2 ML workflow

ML software systems [58] follow ML techniques to find patterns in the data with Software Engineering (SE) standard procedures. ML systems are different from traditional software systems as the behavior is not explicit. There are still concerns about the best practices in ML workflow with the integration of agile methods. For example, Microsoft follows the ML workflow [59] as requirements, data collection, data cleaning, data labeling, feature engineering, model training, model evaluation, model deployment, and model monitoring. In [58] the authors state that ML software systems challenges are quality attributes - adaptability, scalability, safety, and privacy - in terms of data acquisition, training, evaluation, and deployment. Data acquisition has limited, incomplete, and disorganized data problems. Training is time-consuming, and complexity increases between features and predictions. Evaluation with traditional (offline) ML metrics can be inconsistent with the system's performance under production. Deployment of new data can produce bias and latency issues. According to [58], the potential solutions are online A/B ML model experiments, transfer learning, and synthetic data. These solutions are for improving ML software system development. However, there are still open questions about ML embedded in usability evaluation in software systems. These include: which interactions between a user and software systems are required to predict positive overall usability, usability smells, and usability issues, among others.

2.3.3 ML embedded in usability evaluation

ML embedded in usability evaluation emerged from the lack of analytic evidence in usability items' relevance. The conventional approach to evaluating usability often relies on subjective methods, which, while commonly employed, carry inherent limitations. One such limitation is the inherent subjectivity presented in assessments conducted by users and experts which could produce bias. Additionally, they lack the capability for real-time adjustments to tailor different software systems. This can provide feedback essential for enhancing UX. In this context, the limited works found are 1) a new machine learning-based evaluation method [24], 2) a semi-automated framework (EmotiUsing) [2], and 3) usability-user experience (UUX) issues prediction [60].

ML usability evaluation in education

Oztekin et al. [24] developed the first ML method among usability evaluation methods. The user testing in this study included 105 students who completed the UseLearn checklist based on the eLearning system (online biology course). The subjective quantitative answers are the input to predict the eLearning system's overall usability (output variable). The performance criteria were the mean square error (MSE) on the testing set and the correlation between the target and predicted value. The researchers follow the pseudo-Pareto analysis where 43% of the causes will produce 70% of the usability problems. Additionally, the researchers found that the overall usability improves with this methodology and severity index. Also, the best prediction model was the multilayer perceptron (MLP) NN [24]. Following this methodology, the most critical and problematic items were related to personalized learning and insufficient information.

ML usability evaluation in social networks

Souza Santos et al. [2] proposed the semi-automated framework known as EmotiUsing to assess usability evaluation. The data is interaction logs and brain signals captured from sensors. The objective quantitative data were applied to ML models for usability smell detection. The researchers established eleven smells classified into task smells (Laborious Task (LT), Cyclic Task (CTk), Missing Task Feedback, etc.) and action level smells (Undescriptive Element (UEt), Missing Action Feedback (MAF), etc.). The log processing follows a filter of the event types collected (Uniform Resource Locator (URL) change, input change, click, keypress, and scroll) to be part of an episode. Also, usability smell annotations were found by experts through video recordings of the participants. The features for the ML models were event type, duration, total duration, URL, Cascading Style Sheets (CSS) path, Document Object Model (DOM) object, and event-type specific. The ML model will be able to predict tasks-level smells for entire episodes and action-level smells for each event. The experiment [2] used three social network websites with usability issues, Perspective (designed by the researchers), Social-Network, and Love-Social. Fourteen users checked the three websites and followed different tasks (sign up for the social network, add a friend, post a message, etc.). The annotations differ among usability experts, so a main annotator was chosen. The dataset was remodeled into three types of classification problems: binary, multiclass, and multilabel. The researchers concentrated on a binary and multiclass classification problem given the small amount of data and class imbalance. In the binary classification problem, the random forest model obtained the best results for task and action level in terms of accuracy and F1 score. The ML models cannot predict possible usability smells in the multiclass classification problem.

ML usability evaluation in user reviews

Bakiu and Guzman [60] were able to detect usability UUX issues in software and video game user reviews. They applied a collocation algorithm for feature extraction, lexical sentiment analysis, ML models to automatically classify UUX issues/strengths, and information visualization. The ML model chosen was SVM with a popular multilabel solution known as the binary relevance method (BR). The classifier was trained with a manual labeled collection (UUX dimensions) of software reviews. The data input to the classifier was accomplished by the following preprocessing steps: 1) identify and remove stop words (Natural Language Toolkit library) [61], 2) reduce words to grammatical roots (snowball stemming algorithm) [62], 3) transform the text into a bag of words, and 4) best feature selection (Chi-Squared metric). This approach can help developers and UUX designers detect which part of the software application must be corrected or improved.

Chapter 3

Methodology

I proposed a methodology to predict usability based on ML techniques on a software system known as Moodle [63]. For the Moodle course, the data collected is from undergraduate students. The data collected from the students are the user interaction with the Moodle course and usability questionnaires integrated into Moodle. The quantitative objective data is from three tasks (Quiz, Forum, and Assignment). Each one with different user interactions is saved and processed. Data processing calculates features from each task. These are the input in ML models to be trained using a three-fold cross-validation procedure to predict usability scores derived from SUS and UseLearn. SUS and UseLearn usability scores are from quantitative subjective answers. This methodology provides an automatic detection tool to predict usability.

3.1 Methods

3.1.1 Moodle course

Moodle [63] is a LMS for course management, assessment, and learning for K-12 and college students. It was developed in PHP by Martin Dougiamas on 2002, and is composed of different components including Quiz, Forum, Assignment, Questionnaire, etc. The course of this research was hosted in Amazon Web Services (AWS). Figure 3.1 depicts the user interface

of this course in “Topics Format” with labels representing three tasks and questionnaires. The first task is to submit a Quiz. The second task is to submit a paragraph in a simple Forum. The third task is to add a PDF file for the Assignment. At the end of the three tasks, two questionnaires were applied to measure the usability of this software system, the System Usability Scale (SUS), and some questions from UseLearn. The three tasks and the two questionnaires are sequential. The student needs to finish each activity to unlock the next one.

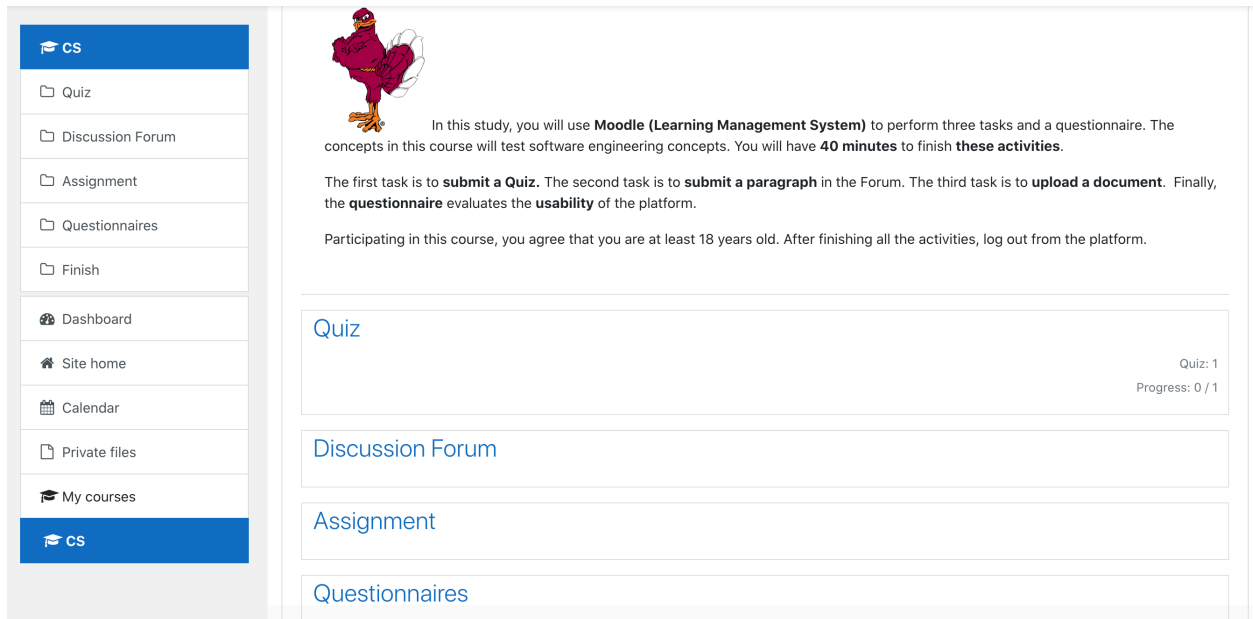
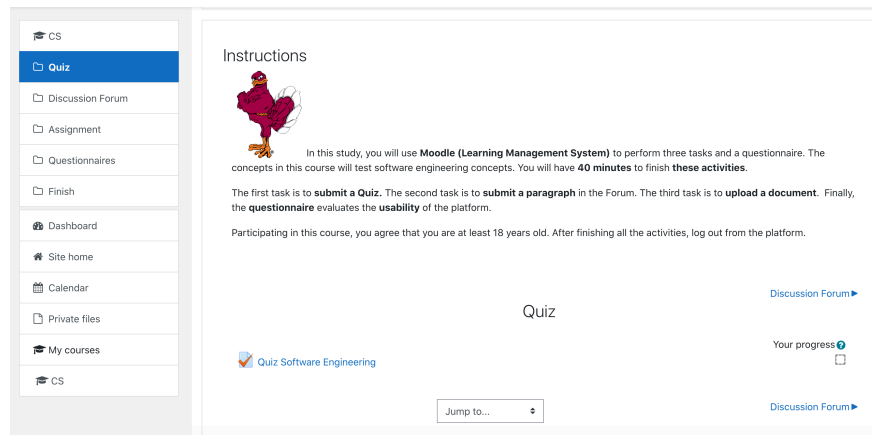


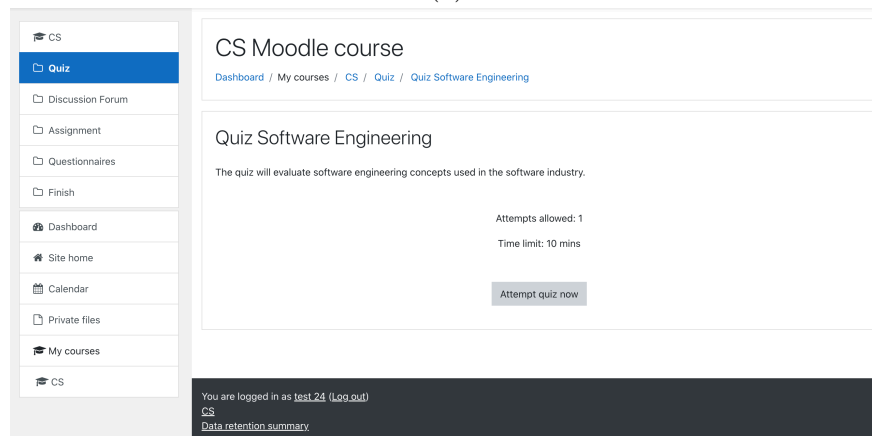
Figure 3.1: Moodle online course.

Quiz

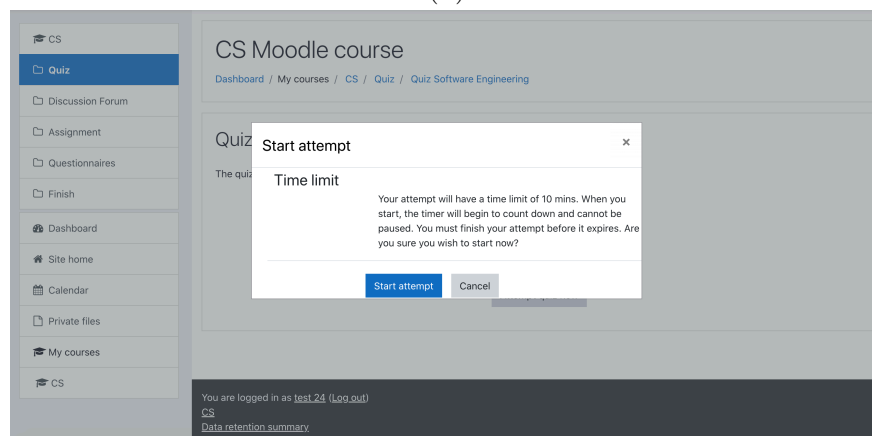
The quiz is composed of 10 shuffled questions about software engineering. The quiz is multiple choice with three possible answers as radio buttons. The quiz layout is consisting of two questions per page with a total of 5 pages. The navigation method is sequential with one attempt allowed. The time constraint is 10 minutes. The steps to finish this task are depicted in Table 3.1. The quiz steps are shown in Figure 3.2 and Figure 3.3.



(a)

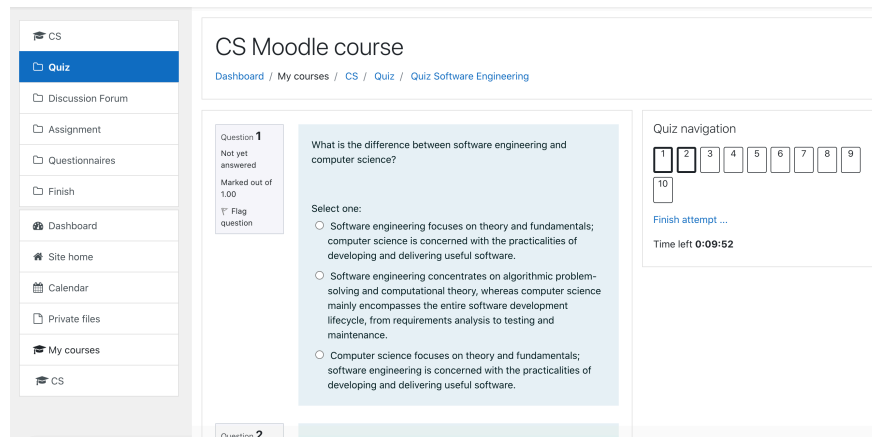


(b)

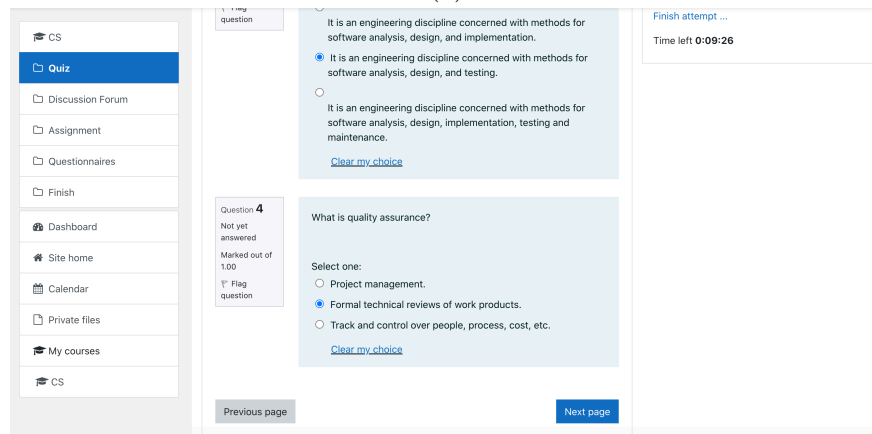


(c)

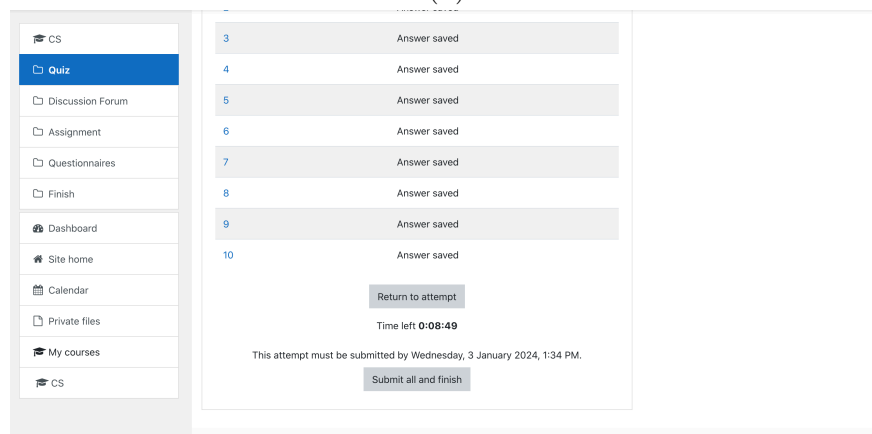
Figure 3.2: Steps 1-3 to finish Quiz. (a) Step 1. Choosing the “Quiz” tab from the left bar from the course, (b) Step 2. Click on “Attempt quiz now”, (c) Step 3. Click on “Start attempt”.



(a)



(b)



(c)

Figure 3.3: Steps 4-6 to finish Quiz. (a) Step 4. Select a “radio button” as an answer, (b) Step 5. Click “Next page” until the 10 questions are answered, (c) Step 6. Click on “Submit all and finish”.

Table 3.1: Steps to finish the quiz.

Step	Information
1	Choosing the “Quiz” tab from the left bar from the course.
2	Click on “Attempt quiz now”.
3	Click on “Start attempt”.
4	Select a “radio button” as an answer.
5	Click “Next page” until the 10 questions are answered.
6	Click on “Submit all and finish”.

Discussion Forum

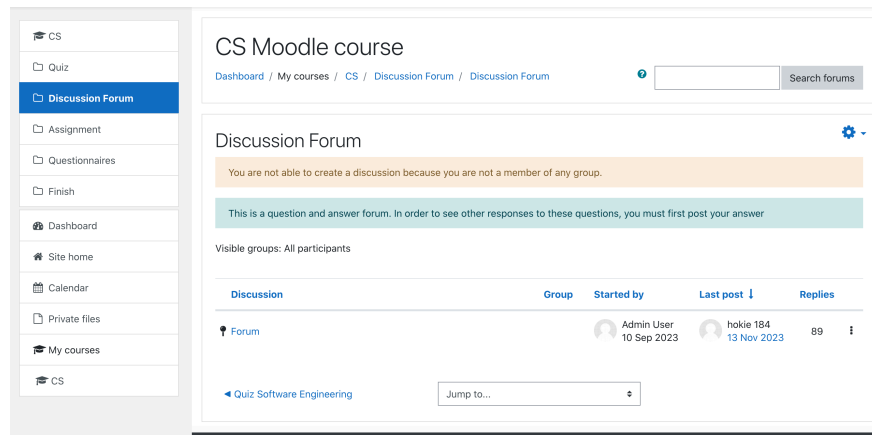
The discussion forum is open to the students based on a question/answer forum basis. The post answer should be relevant to the following statement: “Write a paragraph about the benefits of software engineering in our daily lives”. The student can reply once to this question. Table 3.2 presents the steps to finish this task. Figure 3.4 visualizes the discussion forum steps.

Table 3.2: Steps to finish posting into the discussion forum.

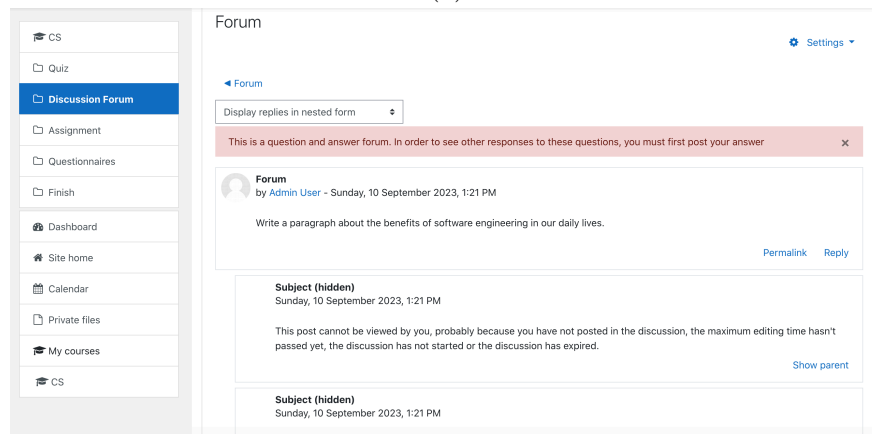
Step	Information
1	Choosing the “Discussion Forum” from the left bar from the course.
2	Click on “Forum”.
3	Click on “Reply” and write the message.
4	Click on “Post to forum”.

Assignment

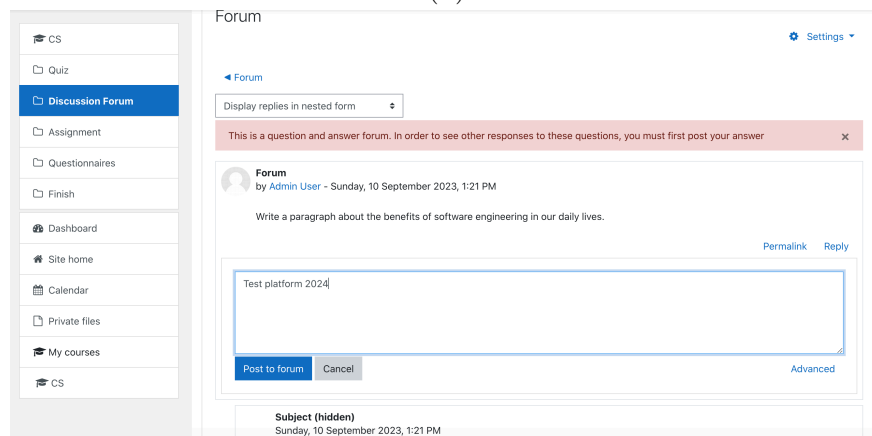
The assignment evaluates a software engineering concept. The student is required to submit a document with a paragraph answering the following question: “What is software engineering?”. The student can submit a maximum of one file and the feedback type is deactivated. The path to finish uploading a file as an assignment in Moodle is shown in Table 3.3. The



(a)



(b)



(c)

Figure 3.4: Steps 1-4 to post into a forum. (a) Steps 1-2. Choosing the “Discussion Forum” from the left bar from the course and click on “Forum”, (b) Step 3. Click on “Reply” and write the message, (c) Step 4. Click on “Post to forum”.

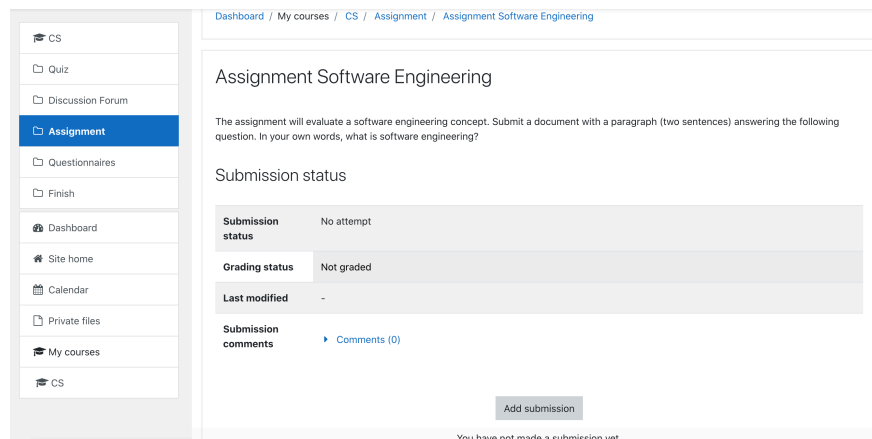
assignment steps are shown in Figure 3.5 and Figure 3.6.

Table 3.3: Steps to finish assignment.

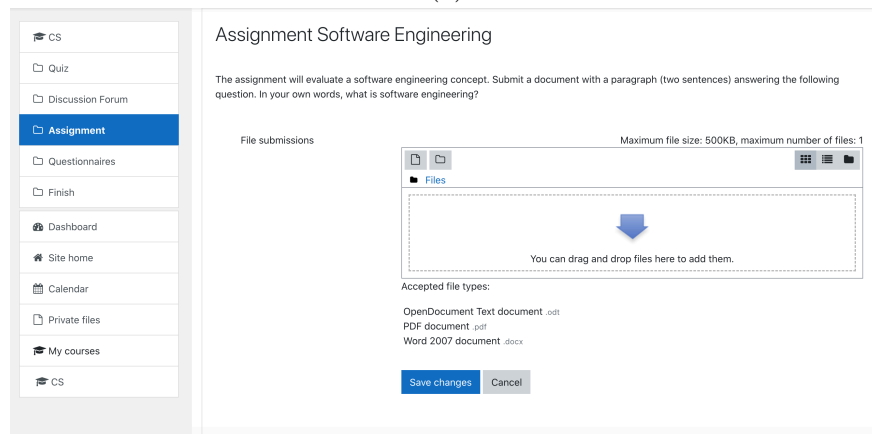
Step	Information
1	Choosing the “Assignment” from the left bar from the course.
2	Click on “Add submission”.
3	Click on the arrow to upload the file.
4	Click on “Choose file”.
5	Click on “Upload this file”.
6	Click on “Save changes”.
7	Click on “Submit assignment”.
8	Click on “Continue”.

3.1.2 Subjective approach

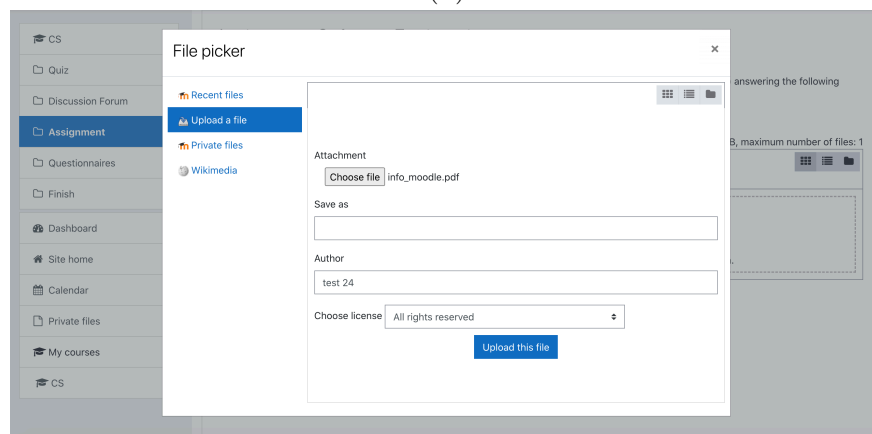
The Moodle course contained questions related to the questionnaires SUS and UseLearn as a subjective approach. The Moodle component was “Questionnaire” and the students can respond to this questionnaire once. They give answers to the SUS [64] for low-level usability dimensions (efficiency, effectiveness, and satisfaction) and some questions from the UseLearn checklist [65] for high-level usability dimensions (error prevention, consistency & functionality, and course management). A change in high usability dimensions will affect the low-level dimensions in the LMS. For example, a button that is not working will directly affect the user perception in terms of efficiency, effectiveness, and satisfaction. SUS is just an overall score about the system’s usability without the reasons behind it. Therefore, a more detailed insight is needed in the high-level dimensions to achieve an efficient and reliable LMS.



(a)



(b)



(c)

Figure 3.5: Steps 1-5 to finish the Assignment. (a) Steps 1-2. Choosing the “Assignment” from the left bar from the course and click on “Add submission”, (b) Step 3. Click on the arrow to upload the file, (c) Steps 4-5. Click on “Choose file” and “Upload this file”.

Assignment Software Engineering

The assignment will evaluate a software engineering concept. Submit a document with a paragraph (two sentences) answering the following question. In your own words, what is software engineering?

File submissions Maximum file size: 500KB, maximum number of files: 1

Files

info_moodle...

Accepted file types:

- OpenDocument Text document .odt
- PDF document .pdf
- Word 2007 document .docx

Save changes Cancel

(a)

Submission status

Submission status	Draft (not submitted)
Grading status	Not graded
Last modified	Wednesday, 3 January 2024, 1:29 PM
File submissions	info_moodle.pdf 3 January 2024, 1:29 PM
Submission comments	Comments (0)

Edit submission Remove submission

You can still make changes to your submission.

Submit assignment

Once this assignment is submitted you will not be able to make any more changes.

(b)

CS Moodle course

Dashboard / My courses / CS / Assignment / Assignment Software Engineering / Confirm submission

Assignment Software Engineering

The assignment will evaluate a software engineering concept. Submit a document with a paragraph (two sentences) answering the following question. In your own words, what is software engineering?

Confirm submission

Are you sure you want to submit your work for grading? You will not be able to make any more changes.

Continue Cancel

Discussion Forum

Jump to...

You are logged in as test24 (Log out)

CS

Data retention summary

(c)

Figure 3.6: Steps 6-8 to finish the Assignment. (a) Step 6. Click on “Save changes”, (b) Step 7. Click on “Submit assignment”, (c) Step 8. Click on “Continue”.

System Usability Scale (SUS)

SUS is a questionnaire with total score ranging from 0 to 100. It is composed of 10 statements with positive and negative questions on a 5-point Likert scale. Scores from each statement will range from 1 (strongly disagree) to 5 (strongly agree). SUS is calculated differently if the questions are even or odd. The score contribution for odd questions is the scale position minus 1. Five minus the score contribution is for even questions. The sum from all the scores is multiplied by 2.5 to obtain the overall value of SUS. The full questionnaire is in Appendix [A.1](#). A score above 70 is considered acceptable.

UseLearn checklist

The UseLearn checklist evaluates specific usability dimensions of the Moodle course. The dimensions of UseLearn are based on 36 questions on a 5-point Likert scale. Scores from each statement will range from 1 (strongly disagree) to 5 (strongly agree). The highest score 5 means that this dimension does not have issues. In this study, three dimensions were selected: 1) error prevention, 2) consistency and functionality, and 3) course management. Each dimension is composed of three questions. These questions are listed in Appendix [A.1](#).

3.1.3 Objective approach

UE based on an objective approach evaluation considers quantifiable metrics on software systems and ML. The metrics are from user interaction with the system (clicks, time, keystrokes, etc.). The user is unaware of these interactions. These interactions and quantitative subjective answers from questionnaires are relevant to ML learning techniques for usability score predictions as an automatic detection tool. Therefore, this section is an overview of the objective strategy used in this study.

1. Log Analysis

Log analysis refers to records collected while the user is using the system. These records [15, 48, 49] collect metrics such as task time, clicks, keystrokes, cursor distance, and completion rate. Figure 3.7 is an example of how these metrics are applied for UE in software systems.

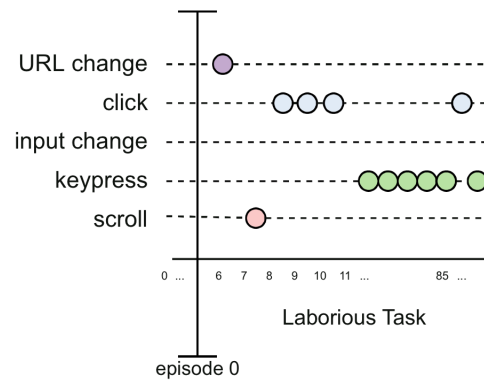


Figure 3.7: Log analysis representation [2].

2. Linear Regression

Linear regression can be univariate or multivariate [3]. In the simple case scenario, univariate means having an input x that fits into a straight line y . The representation is $y = w_1x + w_0$, where w_0 and w_1 are regression coefficients. The term y changes by changing the regression coefficients. It can be defined as $h(x)_w = w_1x + w_0$, where the purpose of linear regression is to find h_w that best fits the data. For example, Figure 3.8 represents a training set of n points, where each point refers to the size and house price. In this case, linear regression is helpful to find a relation between x and y . Least squares are the standard approach [66] to find the regression coefficients that minimize the residual sum of squares between the target and the prediction. In a multivariate linear regression problem, the example x_j is an n -element vector, where the hypothesis space is:

$$h_{sw}(x_j) = w_0 + w_1x_{j,1} + w_2x_{j,2} + \dots + w_nx_{j,n} \quad (3.1)$$

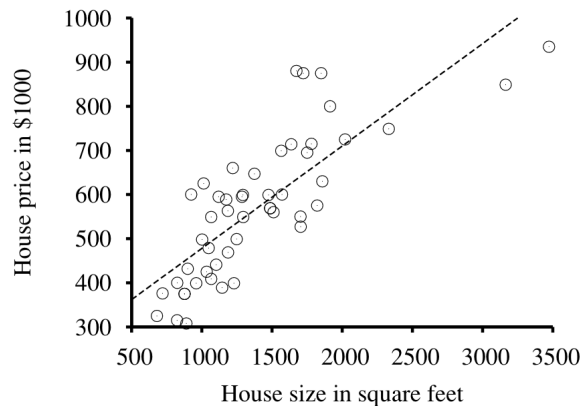


Figure 3.8: Linear regression representation [3].

3. Decision Tree

A decision tree [66] is an algorithm developed by Leo Breiman known as the Classification and Regression Tree (CART). A decision tree [3] is composed of nodes and branches through recursive partitioning. The input attribute corresponds to an internal node (A_i) and the attribute values are in the branches of each node ($A_i=v_{ik}$). The leaves (end branches) are the decision of the tree. For example, Figure 3.9 shows that the attributes are “Body Temperature” and “Gives Birth”. The attribute values for “Body Temperature” are “Warm-blooded” and “Cold-blooded”. The leaves of the tree are “Mammals” and “Non-mammals”. The decision tree chooses the split to minimize the outcome impurity within each sub-partition. The impurity is measured by Gini impurity (classification) [67] or squared deviations from the mean (regression).

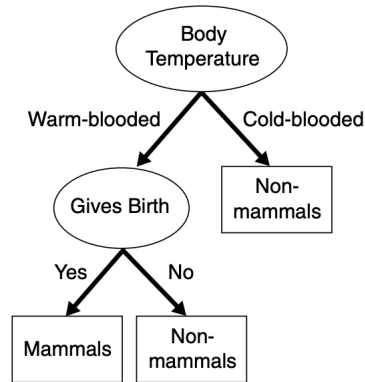


Figure 3.9: Decision tree representation [4].

4. Random Forest

Random forest [4] is a class of ensemble methods (combining multiple individual models) designed for decision trees. Figure 3.10 is an example representation, where each tree training is through a random sample of the training data with replacement (bagging). A random subset of features is applied for the tree splitting. This algorithm [66] chooses the variable and split point by minimizing a criterion such as Gini impurity or squared deviations from the mean. The final prediction is found by averaging the predictions of the individual trees in the forest. Random forest has a set of hyperparameters that should be tuned using cross-validation to avoid overfitting.

5. Artificial Neural Networks

Artificial neural networks [3] are based on the idea of how neurons are connected in the human brain. Figure 3.11 is an instance of a simple neuron with inputs, weights, activation function, and output. A set of neurons creates a graph network, where each node is a neuron connected through a link. The features from the dataset serve as the input in the neural

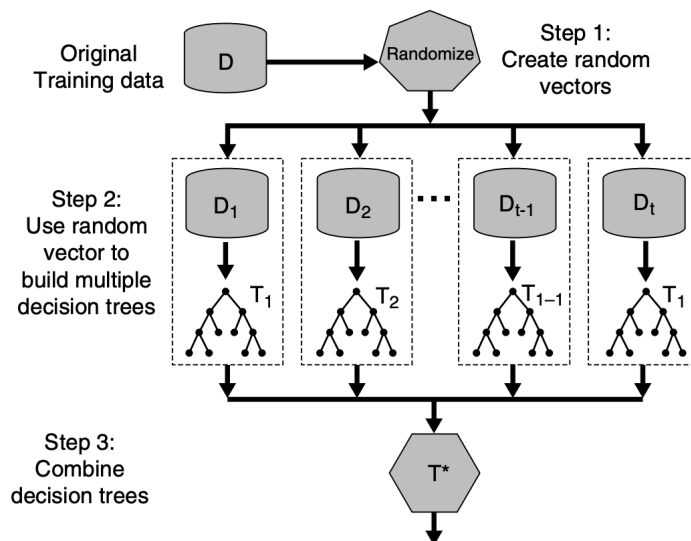


Figure 3.10: Random forest representation [4].

networks to make predictions for classification or regression problems. The architectures are single and multilayer feed-forward neural networks as seen in Figure 3.12. In detail, each node i produces an output with the activation function from the weighted sum of the node's inputs. This output is fed as input to the nodes connected to node i . The learning algorithm is back-propagation, which implements stochastic gradient descent. This is to minimize the loss by iterative updating of the weights in each epoch to converge. The backpropagation error is the difference between the network output and the target.

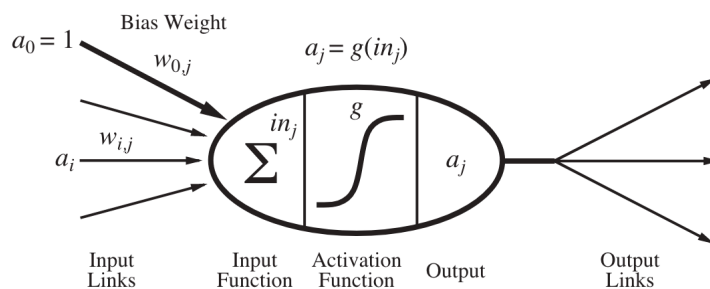


Figure 3.11: Neuron unit representation [3].

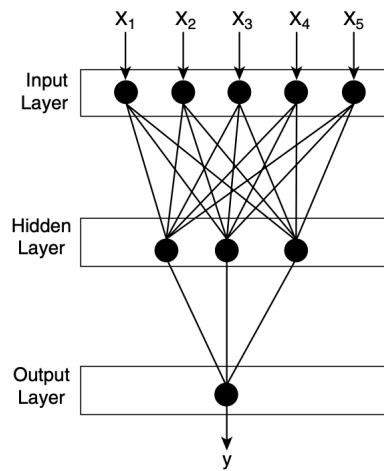


Figure 3.12: Multilayer artificial neural network (ANN) representation [4].

3.2 Experimental Set-Up

3.2.1 Environment

I evaluated the experiments online through Google Collab [68], a Jupyter Notebook service with free access to computing resources. I ran the experiments with the open software Python library scikit-learn [69].

3.2.2 Data collection

We conducted a remote study at Virginia Tech, Blacksburg, Virginia with student participants from undergraduate courses. The participants are Virginia Tech undergraduate students 18 years or older enrolled in an undergraduate course. There are potential benefits for participants in this research such as 1 bonus point in the undergraduate course or being selected to access 1 of the 10 Amazon gift cards in a drawing. At the same time, they could find usability issues in the platform and improve their own learning experience. If participants decide to leave the study for any reason, they are allowed to do so.

I recruited students by contacting different instructors from Virginia Tech by email after the Institutional Review Board (IRB) 23-837 protocol (Appendix A.2.4). The instructors were from the courses CS-2XXX, CS-3XXX, CS-4XXX, BIT-3XXX, BIT-4XXX, ENGE-1XXX, CMDA-36XX, and CMDA-46XX (Appendix A.2.1). A total of 71 emails were sent to faculty members. Fourteen instructors from the Computer Science department replied to the email. They shared the information in their Canvas course and/or permission for a presentation about my research. In total, I did seven presentations of 3 minutes. Additionally, a flyer was posted around campus (Appendix A.2.3).

I shared a Google form survey with a QR code to obtain students' emails (Appendix A.2.2). Afterward, I sent an email to the students about the purpose of the research, log in details, and consent form (Appendix A.2.5). The main goal is to evaluate the usability of the LMS Moodle course with three tasks to check the limitations and difficulties of usability from LMS from the student's perspective. At the end of all tasks, the participants filled out the post-survey SUS and some questions from the UseLearn checklist. Each participant should finish in a maximum of 40 minutes for task completion and post-study questionnaires.

3.2.3 Data pre-processing

Data cleaning

I recruited 183 undergraduate students' emails from Virginia Tech through Google Forms. A total of 88 students logged into Moodle. Eighteen students were excluded. The reasons are logging into the system multiple times, exceeding 40 minutes in the study, uploading unrelated files to the task, posting irrelevant answers to the forum, and not answering the usability questionnaires. Table 3.4 provides a comprehensive overview of the reasons for exclusion. It's worth noting that some students may overlap in terms of the reasons for their

exclusion. Therefore, 70 students were part of the dataset, who finished the three tasks and the usability questionnaires. Forty minutes was the maximum given that the standard time to finish the three tasks and post-surveys was 20 minutes. A student who spent more than 40 minutes did not follow the study guidelines such as finishing the tasks continuously instead of logging into the system multiple times and concentrating on accomplishing these tasks.

Reason	Number of Students
Posting irrelevant answers to the forum	1
Taking more than 40 minutes to finish the tasks	11
Uploading unrelated files to the assignment	4
Logging into the system multiple times	11
Do not answer the usability questionnaires	7

Table 3.4: Exclusion criteria.

Feature extraction

Feature	Type	Description
time quiz	Numerical	seconds to finish quiz task
time forum	Numerical	seconds to finish forum task
time assignment	Numerical	seconds to finish assignment task
grade quiz	Numerical	quiz points (1 correct answer, 0 incorrect answer)
word count per forum response	Numerical	number of words in the post
file size	Numerical	kilobytes in the assignment file
file type	Numerical	1 (PDF file) or 0 (other file type)
clicks module quiz visited	Numerical	number of clicks to finish quiz task
clicks module forum visited	Numerical	number of clicks to finish forum task
clicks module assignment visited	Numerical	number of clicks to finish assignment task

Table 3.5: Features description.

The data was saved in MySQL, a database from Moodle. A sample of the database is seen in Figure 3.13. Algorithms 1 and 2 check task completion and each module table retrieval.

id	eventname	component	action	target	crud	userid	courseid	date	time
12872	\core\event\user_loggedout	core	loggedout	user	r	211	0	11/13/23	10:51:45
12871	\core\event\dashboard_viewed	core	viewed	dashboard	r	211	0	11/13/23	10:51:42
12870	\core\event\course_module_completion_updated	core	updated	course_module_completion	u	211	3	11/13/23	10:51:26
12869	\core\event\course_module_completion_updated	core	updated	course_module_completion	u	211	3	11/13/23	10:51:26
12868	\mod_page\event\course_module_viewed	mod_page	viewed	course_module	r	211	3	11/13/23	10:51:26
12867	\mod_questionnaire\event\attempt_submitted	mod_questionnaire	submitted	attempt	c	211	3	11/13/23	10:51:22
12866	\core\event\course_module_completion_updated	core	updated	course_module_completion	u	211	3	11/13/23	10:51:22
12865	\mod_questionnaire\event\course_module_viewed	mod_questionnaire	viewed	course_module	r	211	3	11/13/23	10:50:23
12864	\mod_assign\event\submission_status_viewed	mod_assign	viewed	submission_status	r	211	3	11/13/23	10:50:20
12863	\mod_assign\event\course_module_viewed	mod_assign	viewed	course_module	r	211	3	11/13/23	10:50:20
12862	\mod_assign\event\assessable_submitted	mod_assign	submitted	assessable	u	211	3	11/13/23	10:50:20
12859	\core\event\course_module_completion_updated	core	updated	course_module_completion	u	211	3	11/13/23	10:50:19
12858	\mod_assign\event\course_module_viewed	mod_assign	viewed	course_module	r	211	3	11/13/23	10:50:19
12857	\mod_assign\event\submission_confirmation_form...	mod_assign	viewed	submission_confirmation_form	r	211	3	11/13/23	10:50:17

Figure 3.13: Database overview.

Algorithm 1 Function to check if the task has been completed

```

1: function IS_COMPLETED(table, componentName, nameActionStart, nameActionEnd,
   nameTargetStart, nameTargetEnd)
2:   componentIncluded  $\leftarrow$  table['component'] == componentName
3:   if componentIncluded.any() then
4:     modVisitedValues  $\leftarrow$  table[table['component'] == componentName]
5:     modVisitedValues  $\leftarrow$  modVisitedValues[['action', 'target']]
6:     startTaskValues  $\leftarrow$  [nameActionStart, nameTargetStart]
7:     endTaskValues  $\leftarrow$  [nameActionEnd, nameTargetEnd]
8:     containsAllStartValues  $\leftarrow$  (modVisitedValues==startTaskValues).all(axis =
   1)
9:     containsAllEndValues  $\leftarrow$  (modVisitedValues==endTaskValues).all(axis =
   1)
10:    if containsAllStartValues.any() and containsAllEndValues.any() then
11:      return True
12:    else
13:      return False
14:    end if
15:  else
16:    return False
17:  end if
18: end function

```

Algorithm 2 Function to get the table from the task completed

```

1: function GET_TASK_COMPLETED_TABLE(userTable, componentName, actionStartName,
   actionEndName, targetStartName, targetEndName)
2:   moduleTable  $\leftarrow$  tableUnique[userTable['component'] == componentName]
3:   indexStart  $\leftarrow$  moduleTable.index[(moduleTable['action'] ==
   actionStartName)&( moduleTable['target'] == targetStartName)].max()
4:   indexStart  $\leftarrow$  moduleTable.loc[indexStart]
5:   indexEnd  $\leftarrow$  moduleTable.index[(moduleTable['action'] ==
   actionEndName)&( moduleTable['target'] == targetEndName)].max()
6:   indexEnd  $\leftarrow$  moduleTable.loc[indexEnd]
7:   moduleTable  $\leftarrow$  moduleTable[(moduleTable['id']  $\leq$ 
   indexEnd.id)&( moduleTable['id']  $\geq$  indexStart.id)]
   return moduleTable
8: end function

```

In Algorithm 1, the input was the original table, component action start, target start, action end, and target end as seen in Table 3.6. For Algorithm 2, the input was each user table with the actions and targets of the respective components, except the session completion component “core”. The module tables extracted from Algorithm 2 were used as input in Algorithms 3 and 4 to find the time and clicks of each module. Finally, based on domain knowledge and prior literature review, the features extracted as input to the supervised ML algorithms are presented in Table 3.5.

Component	Action Start	Action End	Target Start	Target End
mod_quiz	started	submitted	attempt	attempt
mod_forum	viewed	created	course_module	post
mod_assign	viewed	submitted	course_module	assessable
core	loggedin	loggedout	user	user

Table 3.6: Components action and target description.

Algorithm 3 Function to get the time to complete the task

```

1: function GET_TIME_TASK_COMPLETED(moduleTable)
2:   timeTable  $\leftarrow$  moduleTable[time]
3:   time = changeToSeconds(timeTable[0]) – changeToSeconds(totalSeconds[–1])
   return time
4: end function

```

Algorithm 4 Function to get task clicks

```

1: function GET_MODULE_CLICKS(moduleTable)
   return len(moduleTable)
2: end function

```

Target extraction

The targets are four usability scores as seen in Table 3.7: SUS score, error prevention (EP) score, consistency and functionality (CF) score, and course management (CM) score.

Target	Type	Description
SUS score	Numerical	usability score between 0 and 100
EP score	Numerical	usability score between 1 and 5
CF score	Numerical	usability score between 1 and 5
CM score	Numerical	usability score between 1 and 5

Table 3.7: Target description.

Data standardisation

Standardization is a common step in preprocessing data for ML algorithms. This technique produces superior performance in these algorithms by reducing inconsistencies related to different scales of the features. It guarantees better convergence and interpretability. The standardization technique was applied to scale nine features. The features depicted in Figure 3.14 include: time quiz, time forum, time assign, grade quiz, word count per forum response, file size, clicks module quiz, clicks module forum, and clicks module assignment. The formula

to rescale each feature is based on the mean and standard deviation. The formula is:

$$z = \frac{x - \mu}{\sigma} \quad (3.2)$$

where z is the standardized feature, x is the original feature, μ is the mean of the feature, and σ is the standard deviation of the feature.

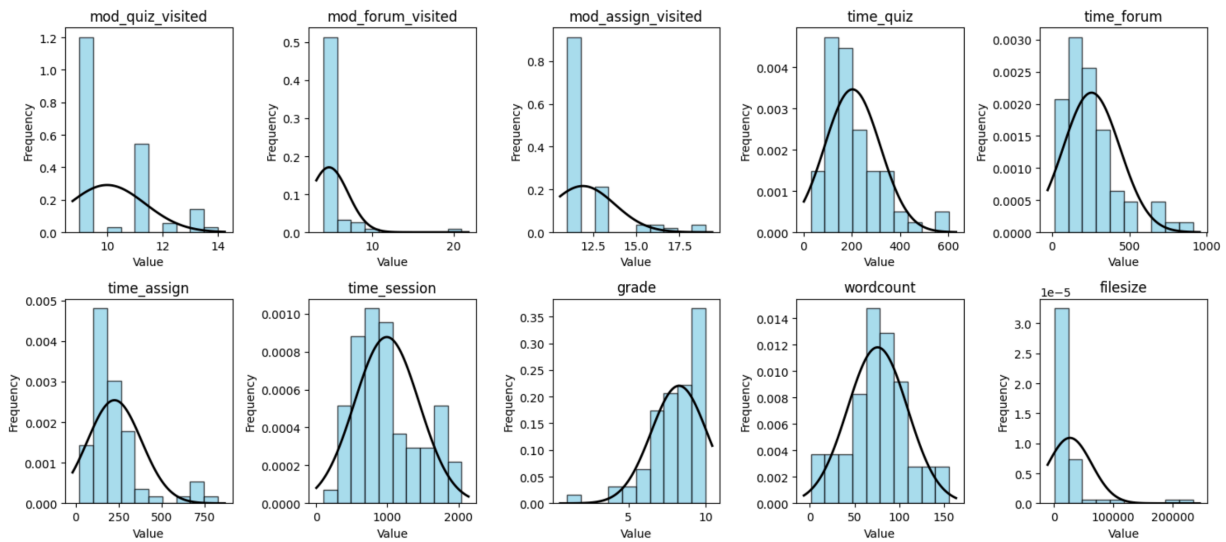


Figure 3.14: Features distribution.

Data normalization

Normalization is a preprocessing technique in ML to ensure consistency and numerical stability between the features (scale between 0 and 1). It enhances overall performance in these algorithms, where the targets are normalized given that the original scale for SUS score is between 0 and 100, and the UseLearn scores between 1 and 5. The formula is:

$$x_{\text{norm}} = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (3.3)$$

where x_{norm} is the normalized target, x is the original target value, $\min(X)$ is the minimum value of the target in the dataset, and $\max(X)$ is the maximum value of the target in the dataset.

3.2.4 Supervised Machine Learning algorithms

I propose a methodology to predict usability based on ML techniques. Three experiments are evaluated depending on the number of features extracted from the three tasks and used as input in the ML algorithms (Table 3.8). The targets are four usability scores: SUS score, EP score, CF score, and CM score. The small dataset collected was applied to ML-supervised algorithms using a three-fold cross-validation procedure to predict the usability scores and avoid overfitting problems. The supervised ML algorithms chosen are linear regression, decision trees, random forest, and neural networks.

Experiment	# Features	Features
1	6	time quiz, time forum, time assignment, clicks module quiz, clicks module forum, and clicks module assignment.
2	7	time quiz, time forum, time assignment, grade quiz, word count per forum response, file size, and file type.
3	10	time quiz, time forum, time assignment, grade quiz, word count per forum response, file size, file type, clicks module quiz, clicks module forum, and clicks module assignment.

Table 3.8: Features description.

The hyperparameters in the ML algorithms were found through the grid search algorithm [70], depicted in Table 3.9.

Table 3.9: Parameter settings for ML algorithms on grid search algorithm.

ML algorithm	Configuration
Linear Regression	fit intercept: [True, False]
Decision Trees	maximum depth: [None, 5, 10, 15, 20, 30], minimum samples leaf: [2, 5, 10], minimum samples split: [2, 5, 10]
Random Forest	maximum depth: [None, 5, 10, 15, 20, 30], minimum samples leaf: [2, 5, 10], minimum samples split: [2, 5, 10], estimators: [10, 50, 100]
Neural Networks	activation: ['relu', 'tanh'], alpha: [0.0001, 0.001, 0.01], hidden layer sizes: [(50), (100), (50,50), (100,50,25), (50,25,10)] learning rate: [0.0001, 0.001, 0.01], solver: ['adam', 'sgd']

3.2.5 Evaluation Metrics

The evaluation metrics are Mean Square Error (MSE) and Root Mean Square Error (RMSE) to verify the performance and fitness in regression models. MSE considers the error as the square difference between the predicted and the actual target. RMSE is the square root of MSE, where the unit of measurement is the same as the target. The formulas are depicted as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m (\hat{Y}_{ij} - Y_{ij})^2 \quad (3.4)$$

$$\text{RMSE} = \sqrt{\text{MSE}} \quad (3.5)$$

where n is the number of records, m is the number of outputs, \hat{Y}_{ij} is the predicted value for the i -th record and j -th output, and Y_{ij} is the target for the i -th record and j -th output.

Chapter 4

Results

This chapter explains the dataset overview about the usability of the Moodle course and the results from the three experiments stated in the Methodology.

4.1 Dataset statistics

Feature	Mean	Standard Deviation
time quiz	203.06	115.02
time forum	255.20	183.41
time assignment	225.54	157.02
grade quiz	8.28	1.81
word count per forum response	75.84	33.77
file size	26735.94	36565.74
file type	0.67	0.47
clicks module quiz	10.00	1.37
clicks module forum	4.71	2.33
clicks module assignment	11.91	1.85

Table 4.1: Features mean and standard deviation.

Table 4.1 shows the mean and standard deviation from the features. It is relevant to remark that the average time to finish the activities was between 203.06 and 255.20 seconds. The average grade quiz was 8, which means that the students understood the questions even without previous learning material. The average clicks to finish the tasks range from 4 to 12 clicks respectively. The average post word count was 76 words related to the open question

about software engineering.

Table 4.2 explains the average and standard deviation of the targets to be predicted by the ML algorithms. The average SUS score is 73, indicating the Moodle course is acceptable in terms of usability. The UseLearn checklist ranges from 3.52 to 3.98 expressing that the Moodle system is usable, but needs improvement in those areas given that 5 is the highest score.

Target	Mean	Standard Deviation	Minimum	Maximum
SUS score	73.11	15.14	27.50	100.00
EP score	3.52	0.72	2.00	5.00
CF score	3.98	0.67	2.67	5.00
CM score	3.81	0.77	1.67	5.00

Table 4.2: Target mean and standard deviation.

The targets are between 0 and 1 after normalization. To be deployed by software developers and UX designers after the usability score prediction, the targets should be transformed to the original range as seen in Table 4.2. For example, a predicted SUS score of 0.7 will be 78.25 using linear scaling.

4.2 Experiments

I proposed three experiments with different features to predict usability scores as a hybrid approach between logging user data, ML, and questionnaires. The insight of these experiments is to show the capacity of supervised ML algorithms to predict usability scores in LMSs and the features involved. Each experiment follows the three cross-validation [71] procedure as a valid strategy to ensure performance in regression problems by the mean and standard deviation of the selected metrics MSE and RMSE [72, 73]. Cross-validation [74] in

a small dataset is a robust metric to estimate model performance. Another consideration is that the research is not a comparative study but a study about which features can predict usability evaluation scores through ML algorithms. The current research about usability and ML uses performance evaluation metrics depending on classification (accuracy and F1 score) or regression (MSE and RMSE) [2, 24, 60]. To demonstrate the ability of ML models to predict usability scores, I considered MSE and RMSE as performance metrics [75].

4.2.1 Experiments with 6 features

Table 4.3: Parameter settings for ML algorithms.

ML algorithm	Configuration
Linear Regression	fit intercept: True
Decision Trees	maximum depth: None, minimum samples leaf: 10, minimum samples split: 2
Random Forest	maximum depth: None, minimum samples leaf: 10, minimum samples split: 2, estimators: 100
Neural Networks	activation: relu, alpha: 0.01, hidden layer sizes: (100, 50, 25) learning rate: 0.001, solver: sgd

In the first experiment, the best ML configuration found by grid search (Table 3.9) is presented in Table 4.3. Based on domain knowledge, I selected six features applicable for usability detection: time quiz, time forum, time assignment, clicks module quiz, clicks module forum, and clicks module assignment.

	SUS score	EP score	CF score	CM score
Linear Regression	0.0576 ± 0.0184	0.0585 ± 0.0028	0.1025 ± 0.0184	0.0698 ± 0.0182
Decision Trees	0.0403 ± 0.0014	0.0603 ± 0.0068	0.1001 ± 0.0197	0.0658 ± 0.0099
Random Forest	0.0363 ± 0.0046	0.0582 ± 0.0106	0.0838 ± 0.0131	0.0550 ± 0.0114
Neural Networks	0.0421 ± 0.0060	0.0589 ± 0.0059	0.0801 ± 0.0125	0.0560 ± 0.0079
Average	0.0441 ± 0.0076	0.0590 ± 0.0065	0.0916 ± 0.0159	0.0617 ± 0.0119

Table 4.4: MSE from different Machine Learning Algorithms with 6 features selected.

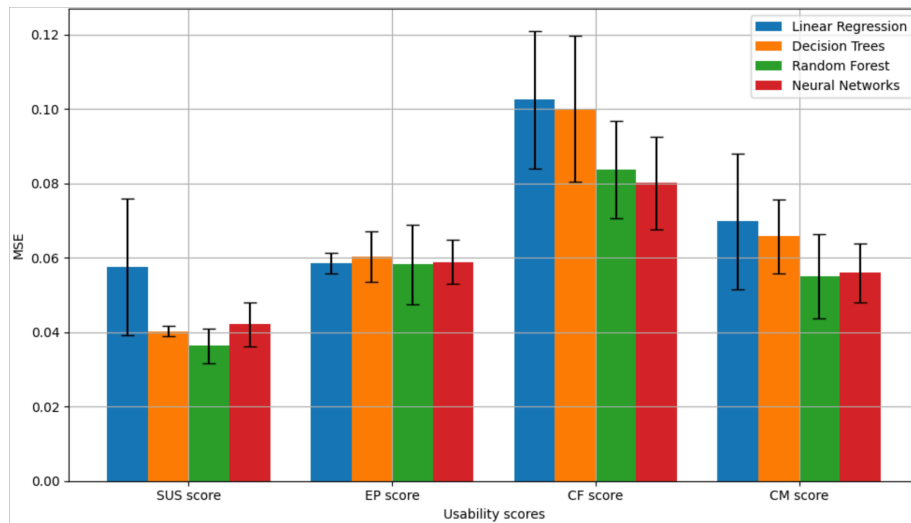


Figure 4.1: MSE from the usability scores with different Machine Learning Algorithms with 6 features selected.

The model prediction results are seen in Figure 4.1 and Table 4.4. The lowest average MSE among the ML algorithms is the SUS score (0.0441 ± 0.0076) and the highest is the CF score (0.0916 ± 0.0159). SUS gives an insight into the overall usability of the software system without specific usability dimensions for LMS. For this reason, the other three scores - CM, EP, and CF - provide usability dimensions specifically for LMS. The UseLearn scores ordered by the average from the ML algorithms are EP score (0.0590 ± 0.0065), CM score

(0.0617 ± 0.0119), and CF score (0.0916 ± 0.0159). Therefore, EP score is helpful to check that multiple tasks are accomplished and with warning preventive error messages. CM score explains that Moodle can provide important information and learning resources, such as uploading files into the system. CF is relevant for consistency between icons, buttons, labels, titles, etc.

	SUS score	EP score	CF score	CM score
Linear Regression	0.2372 ± 0.0369	0.2417 ± 0.0059	0.3188 ± 0.0291	0.2617 ± 0.0360
Decision Trees	0.2008 ± 0.0034	0.2451 ± 0.0137	0.3147 ± 0.0324	0.2557 ± 0.0189
Random Forest	0.1902 ± 0.0122	0.2403 ± 0.0218	0.2887 ± 0.0225	0.2333 ± 0.0236
Neural Networks	0.2046 ± 0.0144	0.2424 ± 0.0123	0.2821 ± 0.0227	0.2361 ± 0.0167
Average	0.2082 ± 0.0167	0.2424 ± 0.0134	0.3011 ± 0.0267	0.2467 ± 0.0238

Table 4.5: RMSE from different Machine Learning Algorithms with 6 features selected.

Random Forest is the best strategy for usability score predictions with overall MSE 0.0583 ± 0.0099 and RMSE 0.2381 ± 0.0200 . The lowest MSE is for Random Forest on the SUS score. Among usability prediction scores, Neural Networks demonstrate notable performance. Linear Regression is the highest MSE and RMSE, meaning that there is no linear relationship between the predictor variables and the target. RMSE proves that in experiment one, Random Forest is the strategy to fit the data to a specific model as shown in Figure 4.2 and Table 4.5. On the other hand, Linear Regression continues to exhibit underperformance, particularly in terms of the CF score.

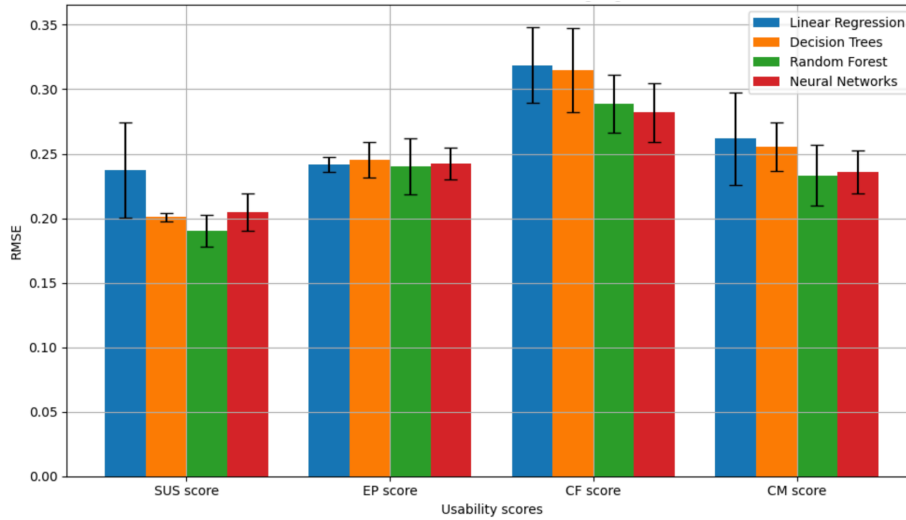


Figure 4.2: RMSE from different Machine Learning Algorithms with 6 features selected.

4.2.2 Experiments with 7 features

In the second experiment, the best ML configuration found by grid search (Table 3.9) is presented in Table 4.6. I deleted the features related to module clicks and added additional features from the specific tasks to verify if the performance improves. The seven features are: time quiz, time forum, time assignment, grade quiz, word count per forum response, file size, and file type.

Table 4.6: Parameter settings for ML algorithms.

ML algorithm	Configuration
Linear Regression	fit intercept: True
Decision Trees	maximum depth: None, minimum samples leaf: 10, minimum samples split: 2
Random Forest	maximum depth: 5, minimum samples leaf: 2, minimum samples split: 5, estimators: 50
Neural Networks	activation: tanh, alpha: 0.01, hidden layer sizes: (100) learning rate: 0.01, solver: sgd

The results from this experiment are seen in Figure 4.3 and Table 4.7. The lowest average

MSE among the ML algorithms is the SUS score (0.0395 ± 0.0055) and the highest is the CF score (0.0867 ± 0.0215). This exhibits the same relevance in those terms as experiment one. The UseLearn scores ordered by the average from the ML algorithms are CM score (0.0599 ± 0.0187), EP score (0.0643 ± 0.0193), and CF score (0.0867 ± 0.0215).

	SUS score	EP score	CF score	CM score
Linear Regression	0.0416 ± 0.0051	0.0649 ± 0.0212	0.0852 ± 0.0209	0.0622 ± 0.0184
Decision Trees	0.0433 ± 0.0014	0.0688 ± 0.0160	0.1028 ± 0.0241	0.0549 ± 0.0127
Random Forest	0.0337 ± 0.0036	0.0599 ± 0.0219	0.0824 ± 0.0209	0.0554 ± 0.0187
Neural Networks	0.0395 ± 0.0118	0.0636 ± 0.0179	0.0765 ± 0.0199	0.0669 ± 0.0250
Average	0.0395 ± 0.0055	0.0643 ± 0.0193	0.0867 ± 0.0215	0.0599 ± 0.0187

Table 4.7: MSE from different Machine Learning Algorithms with 7 features selected.

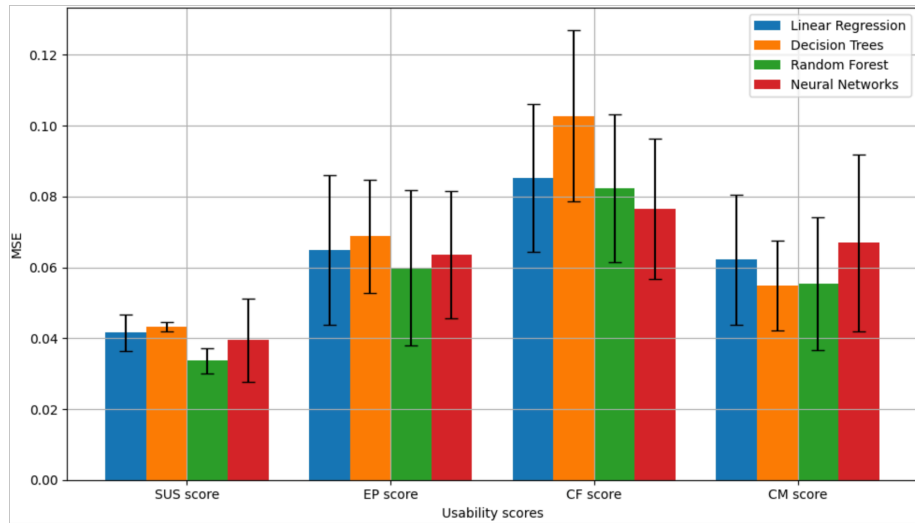


Figure 4.3: MSE from different Machine Learning Algorithms with 7 features selected.

In experiment two, Random Forest obtains the lowest overall MSE (0.0579 ± 0.0162) and RMSE (0.2353 ± 0.032) from all scores. The lowest MSE on Random Forest is 0.0337 ± 0.0036 on the SUS score. Neural Networks are able to predict usability prediction scores as well. On this occasion, Decision Trees demonstrate poor fit to the data, evidenced by their lowest MSE when compared to other techniques. Figure 4.4 and Table 4.8 depict RMSE

from all the ML techniques, where Random Forest has the lowest RMSE results in terms of SUS score (0.1833 ± 0.0099), CM score (0.2323 ± 0.0381), and EP score (0.2410 ± 0.0428).

	SUS score	EP score	CF score	CM score
Linear Regression	0.2037 ± 0.0125	0.2510 ± 0.0432	0.2897 ± 0.0360	0.2463 ± 0.0396
Decision Trees	0.2081 ± 0.0034	0.2606 ± 0.0296	0.3184 ± 0.0375	0.2327 ± 0.0275
Random Forest	0.1833 ± 0.0099	0.2410 ± 0.0428	0.2846 ± 0.0370	0.2323 ± 0.0381
Neural Networks	0.1967 ± 0.0292	0.2497 ± 0.0357	0.2744 ± 0.0349	0.2536 ± 0.0513
Average	0.1980 ± 0.0138	0.2506 ± 0.0378	0.2918 ± 0.0364	0.2412 ± 0.0391

Table 4.8: RMSE from different Machine Learning Algorithms with 7 features selected.

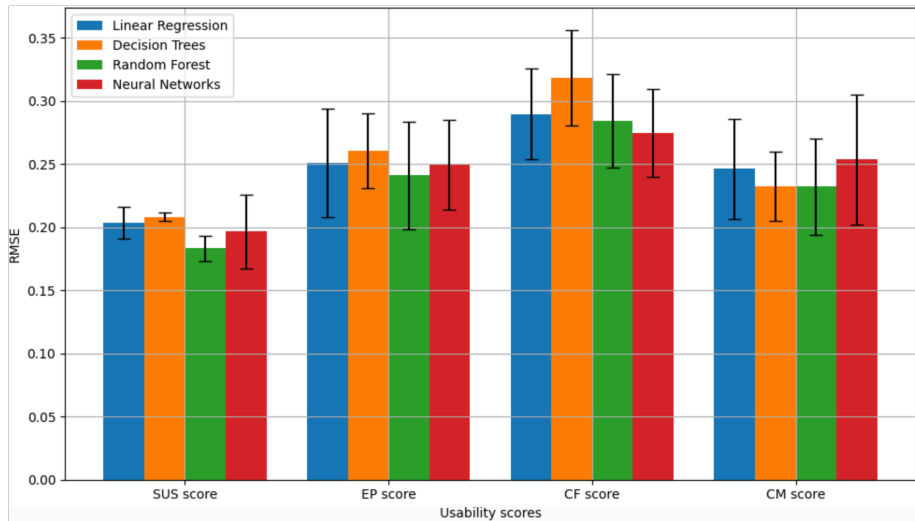


Figure 4.4: RMSE from different Machine Learning Algorithms with 7 features selected.

4.2.3 Experiments with 10 features

In the last experiment, I added the deleted features from experiment two. The ten features in total are time quiz, time forum, time assignment, grade quiz, word count per forum response, file size, file type, clicks module quiz, clicks module forum, and clicks module assignment. These features were selected as input in each algorithm with the configuration presented in Table 4.9.

Table 4.9: Parameter settings for ML algorithms.

ML algorithm	Configuration
Linear Regression	fit intercept: True
Decision Trees	maximum depth: None, minimum samples leaf: 10, minimum samples split: 2
Random Forest	maximum depth: None, minimum samples leaf: 2, minimum samples split: 2, estimators: 50
Neural Networks	activation: relu, alpha: 0.01, hidden layer sizes: (50, 25, 10) learning rate: 0.01, solver: sgd

The model prediction results are seen in Figure 4.5 and Table 4.10. The lowest average MSE among the ML algorithms is the SUS score (0.0456 ± 0.0077) and the highest is the CF score (0.0897 ± 0.0183). The order among the UseLearn scores considering the average from the ML algorithms is CM score (0.0604 ± 0.0176), EP score (0.0691 ± 0.0161), and CF score (0.0897 ± 0.0183).

	SUS score	EP score	CF score	CM score
Linear Regression	0.0599 ± 0.0180	0.0771 ± 0.0118	0.0986 ± 0.0208	0.0652 ± 0.0217
Decision Trees	0.0425 ± 0.0025	0.0662 ± 0.0181	0.1049 ± 0.0217	0.0556 ± 0.0119
Random Forest	0.0364 ± 0.0047	0.0570 ± 0.0196	0.0832 ± 0.0187	0.0540 ± 0.0174
Neural Networks	0.0437 ± 0.0054	0.0759 ± 0.0149	0.0721 ± 0.0118	0.0667 ± 0.0192
Average	0.0456 ± 0.0077	0.0691 ± 0.0161	0.0897 ± 0.0183	0.0604 ± 0.0176

Table 4.10: MSE from different Machine Learning Algorithms with 10 features selected.

	SUS score	EP score	CF score	CM score
Linear Regression	0.2421 ± 0.0355	0.2769 ± 0.0208	0.3122 ± 0.0328	0.2514 ± 0.0444
Decision Trees	0.2061 ± 0.0062	0.2550 ± 0.0340	0.3222 ± 0.0332	0.2344 ± 0.0254
Random Forest	0.1905 ± 0.0121	0.2355 ± 0.0393	0.2865 ± 0.0325	0.2297 ± 0.0358
Neural Networks	0.2086 ± 0.0132	0.2741 ± 0.0281	0.2677 ± 0.0214	0.2553 ± 0.0390
Average	0.2118 ± 0.0168	0.2604 ± 0.0306	0.2972 ± 0.0300	0.2427 ± 0.0362

Table 4.11: RMSE from different Machine Learning Algorithms with 10 features selected.

Random Forest is the best strategy for usability predictions with overall MSE $0.0577 \pm$

0.0151 and RMSE 0.2356 ± 0.0299 . The lowest MSE on Random Forest is 0.0364 ± 0.0047 on the SUS score. Neural Networks excel for usability prediction scores as Random Forest. In this experiment, Linear regression does not fit the data well as experiment one. Figure 4.6 and Table 4.11 depict RMSE from all the ML techniques, where Random Forest has the lowest RMSE results in terms of SUS score (0.1905 ± 0.0121), CM score (0.2297 ± 0.0358), and EP score (0.2355 ± 0.0393).

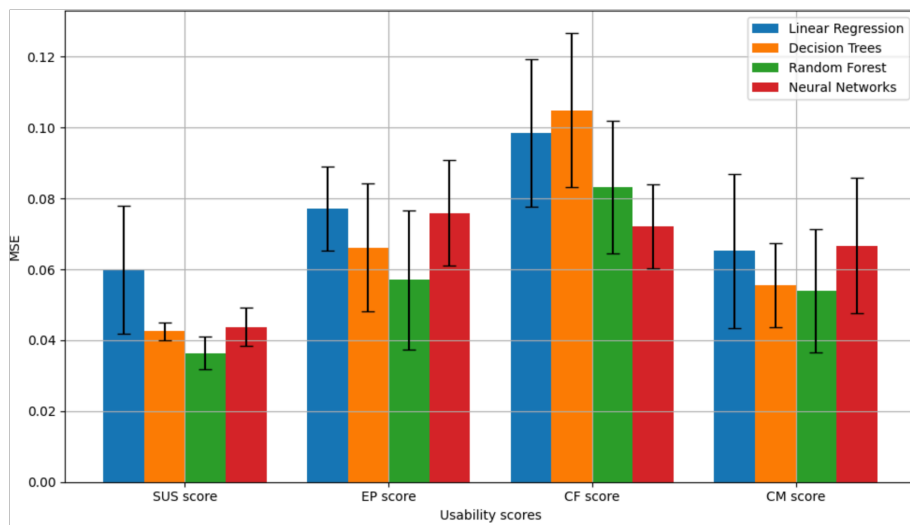


Figure 4.5: MSE from different Machine Learning Algorithms with 10 features selected.

The results of MSE and RMSE show that the seven features provide better results in the SUS score. In this context, there is a potential for usability prediction scores based on quantitative questionnaire answers, user interactions, and ML algorithms.

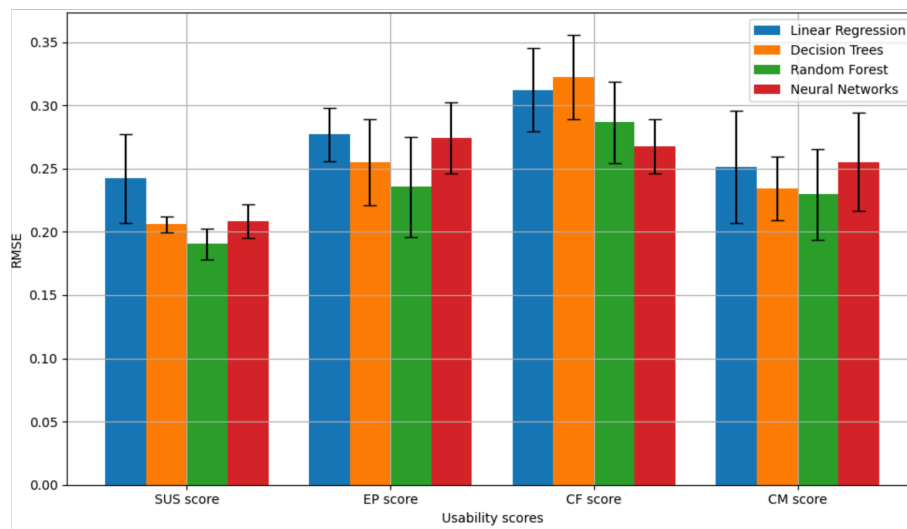


Figure 4.6: RMSE from different Machine Learning Algorithms with 10 features selected.

Chapter 5

Discussion

The results of this study show a prospective adaptable UE alternative for usability score prediction in LMS systems. In the three experiments from subsection 4.2.1 to subsection 4.2.3, ML algorithms obtained the lowest MSE and RMSE of the SUS scores in contrast with the other metrics. SUS score offers software developers and UX designers insight into the system's usability for users (undergraduate students). ML models fit the features more suitable for SUS score prediction (low-level usability dimensions) than the UseLearn checklist (high-level usability dimensions). CF scores perform poorly by the highest RMSE and MSE among the three experiments. This shows that the features selected are not relevant for predicting consistency and functionality between elements in the user interface of the Moodle course, such as consistency between the titles, headers, and icons. The order of UseLearn dimensions is the same in the last two experiments: CM score, EP score, and CF score (Table 4.7 and Table 4.10). ML algorithms have some capability of predicting the CM score, which is relevant for indicating the Moodle course's provision of resources to support online learning. The EP score verifies error prevention measures, ensuring tasks are effortlessly completed. Random Forest achieves the best performance among the experiments. The second experiment of subsection 4.2.2 with seven features produces the best performance of the SUS score. The features are time quiz, time forum, time assignment, grade quiz, word count message post, file size, and file type. The module clicks features from each task were eliminated. This demonstrates that utilizing module clicks is not the best strategy for

feature selection in the SUS score prediction.

ML algorithms with different features exhibit different levels of performance, and require distinct considerations. Linear regression struggles to find a linear relation between the features and the targets. The worst strategy is in the third experiment (subsection 4.2.3), where ten features are selected. Its SUS score improves when module clicks are eliminated, and features of each task are chosen, showing some linear relationship between the inputs and the targets. It is still one of the lowest performances among the three other models. Decision trees enhance RMSE and MSE in experiments one and three (subsection 4.2.1 and subsection 4.2.3). The hyperparameters in the three experiments produce the same values by following a pre-pruning technique (removing unnecessary branches). This depth restricts the number of tree splits from the root node to the leaf nodes. The depth found was “None” meaning that nodes expand until the leaves are pure or the leaves contain less than the minimum samples to split. The value was two samples to split an internal node to capture patterns correlating with the small dataset. The minimum sample leaf is ten to be a leaf node to prevent overfitting with few samples. Decision Trees in experiment two do not produce favorable usability score prediction with the highest RMSE and MSE in some scores. Random Forest is the best algorithm for usability prediction in the three experiments. Random Forest is an ensemble method that aggregates predictions of multiple decision trees as estimators. The rest of the hyperparameters follow the same criteria as an individual tree with the maximum depth, minimum samples split, and minimum samples leaf. In experiment two, the prediction of multiple trees followed criteria with the maximum depth being five. The number of branches decreased, showing the minimum samples for split is five and the minimum samples for leaf is two. As a result, multiple trees produce superior capabilities of RMSE usability prediction for SUS score (0.1833 ± 0.0099), CM score (0.2323 ± 0.0381), and EP score (0.2410 ± 0.0428). Neural networks can predict usability scores in

all the experiments. The second experiment from subsection 4.2.2 can predict CF scores in contrast with the other ML models. The hidden layer is just one layer with 100 neurons. Therefore, neural networks find hidden patterns in the seven features. The learning rates in the experiments range from 0.001 to 0.01. The lowest is in the first experiment where the optimization process converges slowly.

The ML algorithms in this research explain a modest performance, albeit not entirely satisfactory. MSE and RMSE in the usability scores show higher values of 0.0337 ± 0.0036 and 0.1833 ± 0.0099 in all the experiments. A possibility to improve performance is the feature engineering and selection to avoid overfitting (no generalization of unseen samples). The features previously selected were time quiz, time forum, time assignment, grade quiz, word count message post, file size, file type, clicks of module quiz, clicks of module forum, and clicks of module upload file. The features were chosen based on domain knowledge from previous research where user logging data was used [2, 24, 48, 60]. In this context, one common feature is the time of each task which aligns with the results from the experiment of subsection 4.2.2. However, in the previous state-of-the-art works, there is no hybrid methodology where user logging data, quantitative questionnaire answers, and ML are employed for usability score prediction. Another factor is how the module clicks were extracted to be considered as a feature with Algorithm 2 and Table 3.6. A plug-in can be developed and installed on Moodle to extract the module clicks of each task. The features specific to each task in the quiz (grade), forum (word count), and assignment (file size, and file type) are applicable just for usability score predictions on LMS. Features such as erroneous clicks to complete a task can be picked as an alternative.

Erroneous clicks mean which other components the student selected outside the one from the specific task. Linear Regression and Decision Trees make it easier to interpret how a usability score is determined. Random Forests and Neural Networks, although often considered “black

boxes” due to their increased complexity, excel at capturing and characterizing nonlinear relationships and interactions in data. Random Forest produces the highest performance in all the experiments given that multiple trees produce an average between them in terms of RMSE and MSE. Neural Networks show progress in performance, which means that the data forms a nonlinear relationship in the data. The feature extraction and additional data points will be helpful to improve the performance of the ML algorithms, especially Neural Networks.

The limitations of this work are the relatively small size of the dataset used, the case study experiment was virtual, and the generalization to unseen tasks. New data collected from students will increase the performance of the ML algorithms and NN could find hidden patterns with its predictive abilities. A controlled environment in person for the research will be beneficial to analyze student concentration on the tasks without any external stimulus.

Chapter 6

Conclusion and Future work

6.1 Conclusion

I presented a hybrid approach based on user logging data, quantitative usability questionnaire answers, and ML techniques to predict usability scores for an LMS to promote student engagement and satisfaction. The usability scores are from SUS and UseLearn (EP, CF, and CM) questionnaires. We evaluated this approach in a remote case study, with data collected from undergraduate students at Virginia Tech. The 70 students follow three tasks in the Moodle LMS, specifically a quiz, discussion forum, and assignment on topics relevant to software engineering.

I applied different features in the ML algorithms Linear Regression, Decision Trees, Random Forest, and Neural Networks in three experiments. The second experiment of subsection [4.2.2](#) with seven features (time quiz, time forum, time assignment, grade quiz, word count per forum response, file size, and file type) produces the best performance of average MSE and RMSE in the SUS score. Random Forest achieves the best performance among the experiments. The feature module clicks are not suitable for usability SUS score prediction. The results are promising with alternatives for improvements for better performance of SUS and UseLearn scores prediction. The alternative is feature engineering through a plugin on Moodle, additional data points, and tests with unseen tasks.

The SUS score prediction will provide a general insight into the LMS's usefulness. The UE specifically for LMS through UseLearn prediction scores will give information about how well the Moodle course is usable in terms of learning resources (CM), error prevention and task accomplishment (EP), and consistency between the visual elements (CF). In ML software systems, the methodology to test their usability evaluation is through traditional subjective approaches based on questionnaires and user testing. Eventually, UE methods have limitations such as time and monetary investments according to the ML system, and bias embedded in the experiment(s). The research community is more concerned with healthcare and education ML software systems than E-commerce or gaming. The standard questionnaire SUS is the most popular to test usability among all the studies. ML embedded in usability evaluation is scarce. Therefore, our approach provides a potential strategy as a predictive tool for usability scores, which would help researchers, software developers, and UX designers create LMSs that satisfy their users' needs.

6.2 Future work

The module clicks can be extracted differently than Algorithm 2. An alternative is to develop and install a plug-in inside Moodle and extract the module clicks of each task. Another feature can be picked, for example, erroneous clicks. Additional data points in the dataset, new features, more hyperparameters in the ML algorithms, and new tests for unseen tasks would improve the ML performance.

At the same time, the methodology can be extended to other LMSs, for instance, Blackboard or Canvas. The Moodle course initially centered on software engineering concepts, yet can be expanded to other disciplines where LMSs are employed. Usability score prediction would provide customized LMSs according to the users' needs.

Bibliography

- [1] A. Fernandez, E. Insfran, and S. Abrahão, “Usability evaluation methods for the web: A systematic mapping study,” *Information and Software Technology*, vol. 53, no. 8, pp. 789–817, 2011.
- [2] F. de Souza Santos, M. Vinícius Treviso, S. P. Gama, and R. P. de Mattos Fortes, “A framework to semi-automated usability evaluations processing considering users’ emotional aspects,” in *Proceedings of the 24th Human-Computer Interaction International Conference*, Virtual Event, 2022, p. 419–438.
- [3] S. J. Russell, P. Norvig, and E. Davis, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.
- [4] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed. New York, NY, USA: Pearson, 2018.
- [5] R. Hartson and P. S. Pyla, *The UX Book: Agile UX Design for a Quality User Experience*, 2nd ed. Cambridge, MA, USA: Morgan Kaufmann, 2018.
- [6] N. Bevan, J. Carter, and S. Harker, “ISO 9241-11 Revised: What Have We Learnt About Usability Since 1998?” in *Proceedings of the 17th International Human-Computer Interaction Conference*, Los Angeles, CA, USA, 2015, pp. 143–151.
- [7] J. Nielsen, *Usability Engineering*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann, 1994.
- [8] ISO/IEC, “25010:2023 systems and software quality requirements and evaluation,” accessed: 2023-09-30. [Online]. Available: <https://www.iso.org/standard/78176.html>

- [9] R. Torres-Molina and M. Seyam, “The intersection of usability evaluation and machine learning in software systems,” in *Proceedings of the IEEE 5th International Conference on Cognitive Machine Intelligence (CogMI)*, Atlanta, GA, USA, 2023, pp. 122–127.
- [10] A. Holzinger, “Usability engineering methods for software developers,” *Communications of the ACM*, vol. 48, no. 1, pp. 71–74, 2005.
- [11] S. Riihiahho, “Usability Testing,” *The Wiley Handbook of Human Computer Interaction*, vol. 1, pp. 255–275, 2018.
- [12] F. Paz and J. A. Pow-Sang, “Usability Evaluation Methods for Software Development: A Systematic Mapping Review,” in *Proceedings of the 8th International Conference on Advanced Software Engineering and its Applications (ASEA)*, Jeju, South Korea, 2015, pp. 1–4.
- [13] K. Ishaq, F. Rosdi, N. Zin, and A. Abid, “Heuristics and Think-aloud Method for Evaluating the Usability of Game-based Language Learning,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 311–324, 2021.
- [14] K. Abuhlfaia and E. de Quincey, “The usability of e-learning platforms in higher education: a systematic mapping study,” in *Proceedings of the 32nd International BCS Human Computer Interaction Conference*, Belfast, United Kingdom, 2018, pp. 1–13.
- [15] N. Harrati, I. Bouchrika, A. Tari, and A. Ladjailia, “Exploring user satisfaction for e-learning systems via usage-based metrics and system usability scale analysis,” *Computers in Human Behavior*, vol. 61, pp. 463–471, 2016.
- [16] A. C.-P. Anabel Martin-Gonzalez and V. Uc-Cetina, “Usability evaluation of an augmented reality system for teaching euclidean vectors,” *Innovations in Education and Teaching International*, vol. 53, no. 6, pp. 627–636, 2016.

- [17] N. P. I. R. Devy, S. Wibirama, and P. I. Santosa, “Evaluating user experience of English learning interface using user experience questionnaire and system usability scale,” in *Proceedings of the 1st International Conference on Informatics and Computational Sciences*, Semarang, Indonesia, 2017, pp. 101–106.
- [18] K. Abuhlfaia and E. de Quincey, “Evaluating the usability of an e-learning platform within higher education from a student perspective,” in *Proceedings of the 3rd International Conference on Education and E-Learning*, New York, NY, USA, 2020, p. 1–7.
- [19] N. Phongphaew and A. Jiamsanguanwong, “Usability Evaluation on Learning Management System,” in *Proceedings of the 2017 International Conference on Usability and User Experience*, Los Angeles, CA, USA, 2018, pp. 39–48.
- [20] H. Pangestu and M. Karsen, “Evaluation of usability in online learning,” in *Proceedings of the International Conference on Information Management and Technology*, Bandung, Indonesia, 2017, pp. 267–271.
- [21] D. Hariyanto, M. Triyono, and T. Köhler, “Usability evaluation of personalized adaptive e-learning system using USE questionnaire,” *Knowledge Management and E-Learning*, vol. 12, no. 1, pp. 85–105, 2020.
- [22] H. Azami and R. Ibrahim, “Development and evaluation of massive open online course (MOOC) as a supplementary learning tool: An initial study,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, pp. 532–537, 2019.
- [23] O. Korableva, T. Durand, O. Kalimullina, and I. Stepanova, “Usability testing of MOOC: Identifying user interface problems,” in *Proceedings of the 21st International Conference on Enterprise Information Systems*, Crete, Greece, 2019, pp. 468–475.
- [24] A. Oztekin, D. Delen, A. Turkyilmaz, and S. Zaim, “A machine learning-based usability

- evaluation method for eLearning systems,” *Decision Support Systems*, vol. 56, pp. 63–73, 2013.
- [25] A. Alshehri, M. Rutter, and S. Smith, “Assessing the relative importance of an e-learning system’s usability design characteristics based on students’ preferences,” *European Journal of Educational Research*, vol. 8, no. 3, pp. 839–855, 2019.
- [26] V. Franzoni, A. Milani, P. Mengoni, and F. Piccinato, “Artificial intelligence visual metaphors in e-learning interfaces for learning analytics,” *Applied Sciences*, vol. 10, no. 20, pp. 1–25, 2020.
- [27] A. Pensabe-Rodriguez, E. Lopez-Dominguez, Y. Hernandez-Velazquez, S. Dominguez-Isidro, and J. De-la Calleja, “Context-aware mobile learning system: Usability assessment based on a field study,” *Telematics and Informatics*, vol. 48, 2020.
- [28] A. Muhammad, A. Siddique, Q. Naveed, U. Khaliq, A. Aseere, M. Hasan, M. Qureshi, and B. Shehzad, “Evaluating usability of academic websites through a fuzzy analytical hierarchical process,” *Sustainability*, vol. 13, no. 4, pp. 1–22, 2021.
- [29] D. Nariman, “Impact of the interactive e-learning instructions on effectiveness of a programming course,” in *Proceedings of the 14th International Conference on Complex, Intelligent and Software Intensive Systems*, Lodz, Poland, 2021, pp. 588–597.
- [30] J. Dieber and S. Kirrane, “A novel model usability evaluation framework (MUSe) for explainable artificial intelligence,” *Information Fusion*, vol. 81, pp. 143–153, 2022.
- [31] D. Zwakman, D. Pal, and C. Arpnikanondt, “Usability Evaluation of Artificial Intelligence-Based Voice Assistants: The Case of Amazon Alexa,” *SN Computer Science*, vol. 2, no. 1, 2021.

- [32] P. Unberath, H. Prokosch, J. Gründner, M. Erpenbeck, C. Maier, and J. Christoph, “EHR-Independent Predictive Decision Support Architecture Based on OMOP,” *Applied Clinical Informatics*, vol. 11, no. 3, pp. 399–404, 2020.
- [33] M. Bockhacker, H. Syrek, M. Elstermann Von Elster, S. Schmitt, and H. Roehl, “Evaluating Usability of a Touchless Image Viewer in the Operating Room,” *Applied Clinical Informatics*, vol. 11, no. 1, pp. 88–94, 2020.
- [34] Y. Ni, M. Bermudez, S. Kennebeck, S. Liddy-Hicks, and J. Dexheimer, “A real-time automated patient screening system for clinical trials eligibility in an emergency department: Design and evaluation,” *JMIR Medical Informatics*, vol. 7, no. 3, 2019.
- [35] A. Sarkar, J. Wade, A. Swanson, A. Weitlauf, Z. Warren, and N. Sarkar, “A Data-Driven Mobile Application for Efficient, Engaging, and Accurate Screening of ASD in Toddlers,” in *Proceedings of the 12th International Conference Universal Access in Human-Computer Interaction*, Las Vegas, NV, USA, 2018, pp. 560–570.
- [36] A. Gates, S. Guitard, J. Pillay, S. A. Elliott, M. P. Dyson, A. S. Newton, and L. Hartling, “Performance and usability of machine learning for screening in systematic reviews: a comparative evaluation of three tools,” *Systematic Reviews*, vol. 8, no. 1, 2019.
- [37] F. Soboczenski, T. Trikalinos, J. Kuiper, R. Bias, B. Wallace, and I. Marshall, “Machine learning to help researchers evaluate biases in clinical trials: A prospective, randomized user study,” *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, 2019.
- [38] M. Alshammari, R. Anane, and R. Hendley, “Design and usability evaluation of adaptive e-learning systems based on learner knowledge and learning style,” in *Proceedings of the 15th International Conference INTERACT*, Bamberg, Germany, 2015, pp. 584–591.

- [39] J. Gerhard and P. Mayr, “Competing in the e-learning environment-strategies for universities,” in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, Big Island, HI, USA, 2002, pp. 3270–3279.
- [40] M. Alshammari, R. Anane, and R. Hendley, “Usability and effectiveness evaluation of adaptivity in e-learning systems,” in *Proceedings of the International Conference on Human Factors in Computing Systems*, San Jose, CA, USA, 2016, pp. 2984–2991.
- [41] K. Ishaq, F. Rosdi, N. Zin, and A. Abid, “Heuristics and Think-aloud Method for Evaluating the Usability of Game-based Language Learning,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 311–324, 2021.
- [42] B. A. Zardari, Z. Hussain, A. A. Arain, W. H. Rizvi, and M. S. Vighio, “QUEST e-learning portal: applying heuristic evaluation, usability testing and eye tracking,” *Universal Access in the Information Society*, vol. 20, no. 3, 2021.
- [43] B. A. Kumar and M. S. Goundar, “Usability heuristics for mobile learning applications,” *Education and Information Technologies*, vol. 24, no. 2, pp. 1819–1833, 2019.
- [44] D. Quiñones, C. Rusu, and V. Rusu, “A methodology to develop usability/user experience heuristics,” *Computer Standards & Interfaces*, vol. 59, pp. 109–129, 2018.
- [45] L. de Carvalho, Y. Évora, and S. Zem-Mascarenhas, “Assessment of the usability of a digital learning technology prototype for monitoring intracranial pressure,” *Revista Latino-Americana de Enfermagem*, vol. 24, 2016.
- [46] R. Fung, D. Chiu, E. Ko, K. Ho, and P. Lo, “Heuristic usability evaluation of university of hong kong libraries’ mobile website,” *Journal of Academic Librarianship*, vol. 42, no. 5, pp. 581–594, 2016.

- [47] O. Daramola, O. Oladipupo, I. Afolabi, and A. Olopade, "Heuristic evaluation of an institutional e-learning system: A Nigerian case," *International Journal of Emerging Technologies in Learning*, vol. 12, no. 3, pp. 26–42, 2017.
- [48] G. Fenu, M. Marras, and M. Meles, "A learning analytics tool for usability assessment in Moodle environments," *Journal of E-Learning and Knowledge Society*, vol. 13, no. 3, pp. 23–34, 2017.
- [49] J. W. Castro, I. Garnica, and L. A. Rojas, "Automated Tools for Usability Evaluation: A Systematic Mapping Study," in *Proceedings of the 24th Human-Computer Interaction International Conference*, Virtual Event, 2022, pp. 28–46.
- [50] W. Nakamura, E. De Oliveira, and T. Conte, "Usability and user experience evaluation of learning management systems a systematic mapping study," in *Proceedings of the 19th International Conference on Enterprise Information Systems*, Porto, Portugal, 2017, pp. 97–108.
- [51] J. Radianti, T. Majchrzak, J. Fromm, and I. Wohlgenannt, "A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda," *Computers and Education*, vol. 147, 2020.
- [52] J. Lu, M. Schmidt, M. Lee, and R. Huang, "Usability research in educational technology: a state-of-the-art systematic review," *Educational technology research and development*, vol. 70, no. 6, pp. 1951–1992, 2022.
- [53] T. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw Hill, 1997.
- [54] S. Shafiq, A. Mashkoo, C. Mayr-Dorn, and A. Egyed, "A Literature Review of Using Machine Learning in Software Development Life Cycle Stages," *IEEE Access*, vol. 9, pp. 140 896–140 920, 2021.

- [55] P. P. Shinde and S. Shah, “A Review of Machine Learning and Deep Learning Applications,” in *Proceedings of the Fourth International Conference on Computing Communication Control and Automation*, Pune, India, 2018, pp. 1–6.
- [56] S. Das, A. Dey, A. Pal, and N. Roy, “Applications of Artificial Intelligence in Machine Learning: Review and Prospect,” *International Journal of Computer Applications*, vol. 115, no. 9, pp. 31–41, 2015.
- [57] H. Mezouar and A. E. Afia, “A Systematic Literature Review of Machine Learning Applications in Software Engineering,” in *Proceedings of the 5th International Conference on Big Data and Internet of Things*, Rabat, Morocco, 2022, pp. 317–331.
- [58] L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, “Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions,” *Information and Software Technology*, vol. 127, 2020.
- [59] S. Amershi, A. Begel, C. Bird, R. Deline, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, “Software engineering for machine learning: A case study,” in *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*, Montreal, Canada, 2019, pp. 291–300.
- [60] E. Bakiu and E. Guzman, “Which feature is unusable? detecting usability and user experience issues from user reviews,” in *Proceedings of the 25th International Requirements Engineering Conference Workshops (REW)*, Lisbon, Portugal, 2017, pp. 182–187.
- [61] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. Sebastopol, CA, USA: O’Reilly Media, 2009.
- [62] P. Stahl, P. Ljunglof, L. Benzahia, A. Chelli, and A. Aries, “Natural

- language toolkit: Snowball stemmer,” accessed: 2024-01-01. [Online]. Available: https://www.nltk.org/_modules/nltk/stem/snowball.html
- [63] M. Dougiamas, “Moodle,” accessed: 2024-01-01. [Online]. Available: <https://moodle.org/>
- [64] J. Brooke, “SUS: A quick and dirty usability scale,” *Usability Evaluation in Industry*, vol. 189, pp. 189 – 194, 1996.
- [65] A. Oztekin, Z. J. Kong, and O. Uysal, “UseLearn: A novel checklist and usability evaluation method for eLearning systems by criticality metric analysis,” *International Journal of Industrial Ergonomics*, vol. 40, no. 4, pp. 455–469, 2010.
- [66] P. Bruce, A. Bruce, and P. Gedeck, *Practical Statistics for Data Scientists*, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2020.
- [67] S. Suthaharan, *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, 1st ed. New York, NY, USA: Springer, 2016.
- [68] Google, “Google colaboratory,” accessed: 2024-01-01. [Online]. Available: <https://colab.google/>
- [69] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of machine learning research*, vol. 12, pp. 2825–2830, 2011.
- [70] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. MIT Press, 2016.
- [71] I. Tougui, A. Jilbab, and J. E. Mhamdi, “Impact of the choice of cross-validation techniques on the results of machine learning-based diagnostic applications,” *Healthcare Informatics Research*, vol. 27, pp. 189 — 199, 2021.

- [72] O. Rainio, J. Teuvo, and R. Klén, “Evaluation metrics and statistical tests for machine learning,” *Scientific Reports*, vol. 14, no. 1, 2024.
- [73] M. Z. Naser and A. H. Alavi, “Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences,” *Architecture, Structures and Construction*, vol. 3, no. 4, pp. 499–517, 2023.
- [74] S. Raschka, “Model evaluation, model selection, and algorithm selection in machine learning,” *Computing Research Repository*, vol. abs/1811.12808, 2020.
- [75] M. V. Sebt, Y. Sadati-Keneti, M. Rahbari, Z. Gholipour, and H. Mehri, “Regression method in data mining: A systematic literature review,” *Archives of Computational Methods in Engineering*, 2024.

Appendices

Appendix A

User study documents

A.1 Questionnaires

A.1.1 System Usability Scale (5-point Likert scale chart)

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

A.1.2 UseLearn checklist (5-point Likert scale chart)

Error prevention

- Can multiple but similar tasks be done easily?
- Can the user easily undo selections, actions, errors in arrangement or management of items?
- Do error or warning messages prevent possible errors from occurring?

Course management

- Does the course contain important information for the online students and link to support areas?
- Does the course provide specific resources to support online student learning?
- Are files easy to upload?

Consistency and functionality

- Is consistent form and style used for various titles and headers?
- Do the activity, icon, button, label, and links provide clear purpose/intent that matches the tasks?
- Does the interface provide adequate “back” button functionality to return to a previous screen?

A.2 Recruitment strategies

A.2.1 Email sent to professors

Dear Dr. [Professor name],

I am Richard, a master computer science student and teaching assistant in the course CS3704 for Fall 2023. I have been working in the research area of usability and machine learning for an eLearning system with Dr. Mohammed Seyam. I have a proposal that I believe could benefit undergraduate Virginia Tech students and the community of developers and UX designers for faster usability issue detection.

Based on my experience as a teaching assistant in the course CS3704 for 1 year. The Moodle course will be based on concepts based on software engineering. Please, I'd like to request your permission to involve your students (CS3704) in this study. Your collaboration will be invaluable. In this study, the participants use Moodle to perform three tasks. The first task is to submit a Quiz. The second task is to submit a paragraph in a simple Forum. The third task is to add a pdf file for the Assignment. Logs will be saved, and survey(s) will be applied to collect data for further analysis. The total duration of the study will be 40 minutes.

I understand the importance of their academic commitments and will ensure minimal disruption. The research will take place during Fall 2023, remotely. This study could be fundamental for promoting an ML usability automatic detection methodology based on the Moodle data collected.

Thank you, and let me know if you have questions and if the students can be involved in this research.

Best, Richard

Dear Dr. [Professor name],

I am Richard, a master computer science student. I have been working in the research area of usability and machine learning for an eLearning system with Dr. Seyam. I have a proposal that I believe could benefit undergraduate Virginia Tech students and the community of developers and UX designers for faster usability issue detection. Please, if possible can you share the next information in your Canvas course? I am available to give a presentation of 3 minutes as well in your class.

Research proposal: [click here](#). IRB approval: [click here](#)

Canvas students: Hi students! A master student in CS is studying Moodle with machine learning techniques that could improve your e-learning experience. If you are interested, please fill out the Google form.


- Google Form: [click here](#)
- Information Sheet: [click here](#)
- Slides: [click here](#)
- What: complete tasks on Moodle
- Where: remotely
- When: platform will be available from [start date] to [end date]
- How: finish the tasks in a maximum of 40 minutes
- Questions: richardat@vt.edu

Thank you, Richard

A.2.2 Student Google Forms

Recruitment Moodle Study

Hi there, fellow students! We're going to take a closer look at why Moodle might be the better choice for your eLearning journey. Moodle with machine learning learning techniques could improve your e-learning experience. Please, select which course gives you information to be part of this study. You will receive guidelines in your email afterward.

richardat@vt.edu [Switch account](#) 

* Indicates required question

Email *

Record richardat@vt.edu as the email to be included with my response

Which course gives you information to be part of this study? *

CS3704

CS3724

CS3744

Other

A copy of your response will be emailed to richardat@vt.edu.

[Submit](#) [Clear form](#)

This form was created inside of Virginia Tech. [Report Abuse](#)

Google Forms



Recruitment Moodle Study Form

Hi there, fellow students! We're going to take a closer look at why Moodle might be the better choice for your eLearning journey. Moodle with machine learning learning techniques could improve your e-learning experience. Please, select which course gives you information to be part of this study. You will receive guidelines in your email afterward.

richardat@vt.edu [Switch account](#)

* Indicates required question

Email *

Record richardat@vt.edu as the email to be included with my response

Which course gives you information to be part of this study? *

- CS-2114
- CS-3304
- CS-3114
- CS-4804
- CS-4824
- Other

A copy of your response will be emailed to richardat@vt.edu.

Submit

Clear form

This form was created inside of Virginia Tech. [Report Abuse](#)

Google Forms



Recruitment Moodle Study Form

Hi there, fellow students! We're going to take a closer look at why Moodle might be the better choice for your eLearning journey. Moodle with machine learning learning techniques could improve your e-learning experience. Please, select which course gives you information to be part of this study. You will receive guidelines in your email afterward.

richardat@vt.edu [Switch account](#)



* Indicates required question

Email *

Record richardat@vt.edu as the email to be included with my response

Which course gives you information to be part of this study? *

- CS-1064
- CS-1114
- CS-1944
- CS-2104
- CS-2114
- CS-2304
- CS-2505
- CS-3604
- CS-3654
- CS-3714
- CS-4104
- CS-4264
- CS-4624
- Other

A copy of your response will be emailed to richardat@vt.edu.

Submit

Clear form

This form was created inside of Virginia Tech. [Report Abuse](#)

Google Forms




Recruitment Moodle Study Form

Hi there, fellow students! Moodle with machine learning learning techniques could improve your e-learning experience. You will receive guidelines in your email afterward.

- What: complete tasks on Moodle
- Where: remotely
- How: finish the tasks in a maximum of 40 minutes

richardat@vt.edu [Switch account](#)

 Draft saved

* Indicates required question

Email *

Record richardat@vt.edu as the email to be included with my response

Do you like to participate in the study? *

Yes ▼

A copy of your response will be emailed to richardat@vt.edu.

Submit

Clear form

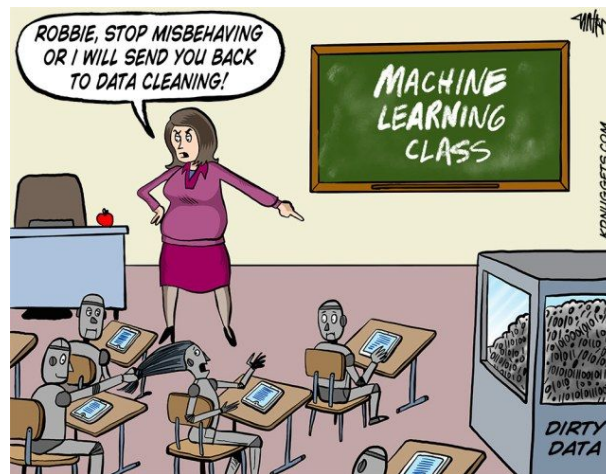
This form was created inside of Virginia Tech. [Report Abuse](#)

Google Forms



A.2.3 Student Flyer

Online education and Machine Learning?



Ready to dive into the fascinating world of online platforms and machine learning? If you're an undergraduate student at Virginia Tech with a curiosity for cutting-edge technology, join our research study as a participant!

No prior knowledge required – just your enthusiasm and willingness to explore. The research will be remotely and you can finish different tasks (survey(s), discussion forum, and upload a file) in a maximum of 40 minutes. You'll enter a draw to receive 1 of 10 Amazon gift cards.

Link: <http://bit.ly/3QdtcUy>

#IRB: 23-837

Scan the QR Code and sign up to send the login details.

Email: richardat@vt.edu

Thank you!



A.2.4 Institutional Review Board (IRB) Confirmation Approval



Division of Scholarly Integrity and
Research Compliance
Institutional Review Board
North End Center, Suite 4120 (MC 0497)
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-3732
irb@vt.edu
<http://www.research.vt.edu/sirc/hrpp>

MEMORANDUM

DATE: October 10, 2023
TO: Mohammed Saad Mohamed Elmahdy Seyam, Richard Andres Torres Molina
FROM: Virginia Tech Institutional Review Board (FWA00000572)
PROTOCOL TITLE: Usability Evaluation of a Learning Management System (LMS): Moodle Case Study
IRB NUMBER: 23-837

Effective October 10, 2023, the Virginia Tech Human Research Protection Program (HRPP) determined that this protocol meets the criteria for exemption from IRB review under 45 CFR 46.104 (d) category(ies) 2(ii).

Ongoing IRB review and approval by this organization is not required. This determination applies only to the activities described in the IRB submission and does not apply should any changes be made. If changes are made and there are questions about whether these activities impact the exempt determination, please submit an amendment to the HRPP for a determination.

This exempt determination does not apply to any collaborating institution(s). The Virginia Tech HRPP and IRB cannot provide an exemption that overrides the jurisdiction of a local IRB or other institutional mechanism for determining exemptions.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

<https://secure.research.vt.edu/external/irb/responsibilities.htm>

(Please review responsibilities before beginning your research.)

PROTOCOL INFORMATION:

Determined As: **Exempt, under 45 CFR 46.104(d) category(ies) 2(ii)**
 Protocol Determination Date: **September 7, 2023**

ASSOCIATED FUNDING:

The table on the following page indicates whether grant proposals are related to this protocol, and which of the listed proposals, if any, have been compared to this protocol, if required.

Invent the Future

A.2.5 Student email confirmation study

Hello,

Thank you for your participation, you will complete three tasks and a post-survey. As part of the study, you will use a LMS (Learning Management System) known as Moodle to perform three tasks related to software engineering concepts. The first task is to submit a survey. The second task is to submit a paragraph in a forum. The third task is to upload a document. Logs will be saved, and a post-survey will be applied to collect data for further analysis. The platform will be available until [end date].

After logging in you have a maximum of 40 minutes to finish the activities mentioned above. Please, log out from the platform after finishing the activities. You'll enter a draw to receive 1 of 10 Amazon gift cards.

- Moodle: [click here](#)
- Username: [username]
- Password: [password]
- Information Sheet: [click here](#)
- Questions: richardat@vt.edu

Best,